

Space functions of groups.

A.Yu.Olshanskii *

November 5, 2010

1 Introduction

Time and space complexities are the main properties of algorithms. Their counterparts in Group Theory are the Dehn and filling length (or space) functions of finitely presented groups. In this paper, we study the interrelation of space functions of groups and the space complexity of the algorithmic word problem in groups.

Let $G = \langle A \mid R \rangle$ be a group presentation, where A is a set of generators and R is a set of defining relators. Recall that relators belong to the free group with basis A , and a group word w in generators A (i.e., a word over $A^{\pm 1}$) represents the identity of G iff there is a rewriting

$$w \equiv w_0 \rightarrow w_1 \rightarrow \cdots \rightarrow w_{t-1} \rightarrow w_t \equiv 1 \quad (1.1)$$

where 1 is the empty word, the sign \equiv is used for the letter-by-letter equality of words, and for every $i = 1, \dots, t$, the word w_i results from w_{i-1} after application of one of the elementary R -transformations. As such transformations one can take free reductions of subwords $aa^{-1} \rightarrow 1$ ($a \in A^{\pm 1}$), removing subwords $r^{\pm 1}$, where $r \in R$, and the inverse transformations.

The minimal non-decreasing function $f(n): \mathbb{N} \rightarrow \mathbb{N}$ such that for every word w vanishing in G and having length $\|w\| \leq n$, there exists a rewriting (1.1) with $t \leq f(n)$, is called the Dehn function of the presentation $G = \langle A \mid R \rangle$ [13]. For *finitely presented* groups (i.e., both sets A and R are finite) Dehn functions are usually taken up to equivalence to get rid of the dependence of a finite presentation for G (see [16]). To introduce this equivalence \sim , we write $f \preceq g$ if there is a positive integer c such that

$$f(n) \leq cg(cn) + cn \quad \text{for any } n \in \mathbb{N} \quad (1.2)$$

For example, we say that a function f is *polynomial* if $f \preceq g$ for a polynomial g . From now we use the following equivalence for non-decreasing functions f and g on \mathbb{N} .

$$f \sim g \quad \text{if both } f \preceq g \text{ and } g \preceq f \quad (1.3)$$

It is not difficult to see that the Dehn function $f(n)$ of a finitely presented group G is recursive (or bounded from above by a recursive function) iff the word problem is algorithmically decidable for G (see [11], [6]). In this case, the word problem can be

*The author was supported in part by the NSF grant DMS 0700811 and by the Russian Fund for Basic Research grant 08-01-00573

solved by a primitive algorithm that, given a word w of length n , just checks if there exists a sequence (1.1) of length $\leq f(n)$. Therefore the nondeterministic *time* complexity of the word problem in G is bounded from above by $f(n)$. Moreover if H is a finitely generated subgroup of G , then one can use the rewriting procedure (1.1) for H , and so the nondeterministic time complexity of the word problem for H is also bounded by $f(n)$.

It turns out that a converse statement is also true. Assume that the word problem can be solved in a *finitely generated* group H by a nondeterministic Turing machine (NTM) with time function $T(n)$. Then G is a subgroup of a *finitely presented* group G with Dehn function equivalent to $n^2T(n^2)^4$ [4]. As the main corollary, one concludes that the word problem of a finitely generated group H has time complexity of class NP (i.e., there *exists* a non-deterministic algorithm of polynomial time complexity, that solves the word problem for H) iff H is a subgroup of a finitely presented group with polynomial Dehn function.

We want to obtain a similar statements for space functions. It is clear that to perform the rewriting (1.1) one needs the space equal to $\max_{0 \leq i \leq t} \|w_i\|$, and this observation leads to the definition of space function of a finitely presented group. However when handling groups, one can enlarge the set of elementary R -transformations and obtain different definitions of space functions. One can either consider only the transformations we defined above and obtain the filling length functions introduced by Gromov [13] (also see [12], [3]), or one can also allow to replace words by their cyclic permutations as it was suggested by Bridson and Riley [7], or can add the replacement of a word $w \equiv uv$ by the pair (u, v) if both u and v are trivial in G (see [7] again). Starting with this different sets of transformations one comes to different space functions called in [7], respectively, filling length function (FL), free filling length functions (FFL), and fragmenting free filling length functions ($FFFL$). Each of these functions has a visual geometric interpretation in terms of the transformations of loops in the Cayley complex of G using, respectively, null-homotopy, free null-homotopy, and free null-homotopy with bifurcations. It is proved in [7] that these functions behave differently for the same finitely presented group G , for instance, $FFFL$ can grow linearly while FL and FFL have exponential growth. There are many other features of these functions presented in [7] to justify their "inclusion in the pantheon of filling invariants".

In the paper, we choose the third version ($FFFL$), and this choice is justified by the theorems on the connection of such functions to the space complexity of the word problem for groups.¹ Thus we operate with finite sequences of words $W = (w_1, \dots, w_s)$ over a group alphabet A . Given a finitely presented group G , we say that a finite sequence $W' = (w'_1, \dots, w'_{s'})$ results from W after application of an elementary R -transformation if $s' \in \{s-1, s, s+1\}$ and one of the following is done for some w_i ($i = 1, \dots, s$):

- a subword aa^{-1} is removed from or inserted to w_i ($a \in A^{\pm 1}$);
- a subword r or r^{-1} is removed from or inserted to w_i ($r \in R$);
- w_i is replaced by a cyclic conjugate;
- $w_i \equiv uv$, and w_i is replaced by the pair u, v , i.e. $W' = (w_1, \dots, w_{i-1}, u, v, w_{i+1}, \dots, w_s)$;
- w_i is removed if it is empty, i.e. $W' = (w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_s)$.

¹An embedding statement for the FL -functions was conjectured by J.-C. Birget in [3].

Clearly, we have $w = 1$ in the group G iff there exists an R -rewriting starting with (w) and ending with the empty string $(\)$.

For every finite sequence $W = (w_1, \dots, w_s)$ we set $\|W\| = \sum_{i=1}^s \|w_i\|$. By definition, the space of a rewriting $W_0 \rightarrow \dots \rightarrow W_t$ is $\max_{j=0}^t \|W_j\|$. If a word vanishes in G , then $space(w) = space_G(w)$ is the minimum of spaces of all rewritings starting with (w) and ending with the empty string. The *space function* of the group presentation $G = \langle A \mid R \rangle$ (or briefly, of the group G) is the function

$$S_G(n) = \max(space(w), \text{ where } w = 1 \text{ in } G \text{ and } \|w\| \leq n)$$

The space functions of finitely presented groups will be regarded up to the equivalence defined by (1.2) and (1.3), and so their growth will be at least linear. It is observed in [7] that the equivalence class of S_G does not depend on a finite presentation of the group G , moreover this class is invariant under quasi-isometries.

An accurate definition of the space function $f(n)$ for a Turing machine (TM) will be recalled in Subsection 2.1. Now we just note that it is usual that for a multi-tape TM , the function $f(n)$ counts only the space of work tapes. However since the space functions of machines are taken here up to the same equivalence as the space functions of groups, the adding of the space of the input tape does not change the equivalence class.

The sequence $W_0 \rightarrow \dots \rightarrow W_t$ can be easily produced by an NTM such that the computation needs at most $2 \max_{i=1}^t \|W_i\| + const$ tape squares. (See also Section 3 of [25] or Remark 2.4 in [7].) This immediately implies

Proposition 1.1. *The space function of a finitely presented group G is equivalent to the space function of a non-deterministic two-tape TM . The language accepted by this machine coincides with the set of words equal to 1 in the group.*

In particular, the non-deterministic space complexity of the word problem in a finitely presented group G does not exceed the space function of G . It follows from [16], [8] that the literally converse statement fails. Moreover, a counter-example can be given by Baumslag's 1-relator group [1] $G = \langle a, b \mid (aba^{-1})b(aba^{-1})^{-1} = b^2 \rangle$ because the space function of G is not bounded from above by any multi-exponential function (see papers of S.Gersten [10] and A.Platonov [23]) while the space complexity of the word problem for G is at most exponential as this was proved by M. Kapovich and Schupp (unpublished), moreover, it is polynomial (announced by A. G. Miasnikov, A. Ushakov, and Dong Wook Won). The correct formulation has to take into consideration that the algorithm from Proposition 1.1 solves the word problem not only for G but also for every finitely generated subgroup of the group G . In the deterministic case we get a sharper formulation:

Theorem 1.2. *Let H be a finitely generated group such that the word problem for H is decidable by a deterministic TM (DTM) with space function $f(n)$. Then H is a subgroup of a finitely presented group G with space function equivalent to $f(n)$.*

The main corollary of this theorem applies to polynomial space complexity. We say that a finitely generated group G belongs to the class $PSPACE$ (to $NPSPACE$) if the word problem for G is decidable by some DTM (some NTM) with a polynomial space function. But $NPSPACE = PSPACE$ by remarkable Savitch's theorem (see [9], Corollary 1.31) which contrasts with deterministic *time* complexity having therefore no natural algebraic counterpart. Proposition 1.1 and Theorem 1.2 eliminate any non-determinism in

Corollary 1.3. *A finitely generated group H belongs to $PSPACE$ iff H is a subgroup of a finitely presented group G having polynomial space function.*

Thus, given a 'good' algorithm solving the word problem in H (e.g., using a matrix representation of H , etc.), it is possible to find a bigger group G whose deterministically modified ('silly') natural algorithm solves the word problem for both H and G , and whose space function is not much worse than the space function of the original algorithm.

Another natural question raised in our paper is the realization problem: Which functions $f(n): \mathbb{N} \rightarrow \mathbb{N}$ are, up to equivalence, the space functions of finitely presented groups? There are not many examples; linear and exponential ones can be found in [7], but it is not easy even to point out a group with space function n^2 .

Theorem 1.4. *Every space function $f(n)$ of a DTM is equivalent to a space function of some finitely presented group.*

This theorem gives a tremendous class of space functions for groups, including functions equivalent to $[\exp \sqrt{n}]$, $[n^k]$ ($k \in \mathbb{N}$), $[n^k \log^l n]$, $[n^k \log^l(\log \log n)^m]$, etc.

The main theorem implies a non-deterministic corollary. To formulate it we recall that a function $s: \mathbb{N} \rightarrow \mathbb{N}$ is called *fully space-constructible (FSC)* if there exists a two-tape DTM that on any input x of length n halts visiting exactly $s(n)$ tape squares of the work tape. Most common functions are FSC (see [9]).

Corollary 1.5. *Let H be a finitely generated group such that the word problem for H is decidable by an NTM having an FSC space function $f(n)$. Then H is a subgroup of a finitely presented group G with space function equivalent to $f(n)^2$.*

Finally, we describe the functions n^α which are (up to equivalence) the space functions of groups. For this aid, we modify the proof of Savitch's theorem from [9] and the approach from [25], where the similar problem was considered for Dehn functions if $\alpha \geq 4$, and close necessary and sufficient conditions were obtained. (See also a dense series of examples with $\alpha \geq 2$ presented by Brady and Bridson [5].) Now we have $\alpha \geq 1$ in Corollary 1.6 below. Also it is remarkable that for space functions the necessary and sufficient conditions just coincide. To formulate the criterion, we call a real number α *computable with space* $\leq f(m)$, if there exists a DTM which, given a natural number m , computes a binary rational approximation of α with an error $O(2^{-m})$, and the space of this computation $\leq f(m)$.

We have got the following criterion.

Corollary 1.6. *For a real number $\alpha \geq 1$, the function $[n^\alpha]$ is equivalent to the space function of a finitely presented group iff α is computable with space $\leq 2^{2^m}$.*

It follows that functions $[n^\pi]$, $[n^{\sqrt{e}}]$, and $[n^\alpha]$ with any algebraic $\alpha \geq 1$ are all the space functions of finitely presented groups.

The space function is defined for a simply connected geodesic metric space under some weak restrictions, in particular, for the universal cover of any closed connected Riemannian manifold. (See [7] for details; we just note here that to define the space (= FFFL) function, one should consider free homotopy with possibility of separating of a loop in two loops in a bifurcation point.) It is proved in [7] (Theorem E) that if a finitely presented group G acts properly and cocompactly by isometries on such a space X , then the space function of G is equivalent to the space function of X . Since every finitely

presented group is a fundamental group of a connected compact Riemannian manifold, we can use corollaries 1.4 and 1.6 and formulate one more

Corollary 1.7. *For every space function $f(n)$ of a DTM, there exists a closed connected and simply connected Riemannian manifold M (with a properly cocompact action of a finitely presented group on it) such that the space function of M is equivalent to $f(n)$.*

In particular, if a real number $\alpha \geq 1$ is computable with space $\leq 2^{2^m}$, then there exists such a manifold with space function equivalent to n^α .

To some extent, our constructions can be traced back to the works of P. Novikov, Boon, Britton and other authors who invented group-theoretical interpretation of TM (see [24], ch. 12). The hub relation copies the accept configuration of a machine several times. Using the language of van Kampen diagram, we correspond to every computation, a disc surrounding the hub cell, and this disc has a number of similar sectors. First of all in the present paper, we are to estimate the sizes of computational discs, and to do this one should know the *generalized* space function of a machine which estimates the space of computations starting with *arbitrary* accept configuration, not only with input ones. So we should modify the initial machine to be able to control the generalized space function. (The modification from [25] helps to control the time function but corrupts the space function.)

The next modification is due to the symmetry of algebraic relations: since $u = v$ always implies $v = u$, the algebraic version of a machine M always interprets the symmetrization of M . Thus we concern that the symmetrization preserves the basic characteristics, e.g. the accepted language and the space functions. We are able to do this only if the initial machine is deterministic or can be transformed to a deterministic under the control of basic properties. (The known symmetrization trick from [2] or [25] does not work here since it does not preserve the space function.) This causes the restrictions in formulations of Theorem 1.2 and Corollary 1.5.

The interpretation problem for groups remains much harder than for semigroups even after the adaptation of the machine because the group theoretic simulation can execute unforeseen computations with non-positive words. Boon and Novikov secured the positiveness of admissible configurations in discs with help of an additional 'quadratic letter' (see [24], ch.12), but this involves a difficult control of parameters for the constructed group. A new approach was suggested in [25]. Invented by Sapir S -machines can work with non-positive words on the tapes. Here we also construct an S -machine which is a somewhat modified composition of a convenient Turing machine with an 'adding machine' $Z(A)$ introduced in [22]. Fortunately, $Z(A)$ does not change the space of computations but controls positiveness of configurations.

Recall that we should not just simulate the work of a machine but construct an embedding of given group H into a finitely presented group G in the spirit of the Higman Embedding Theorem. (see [24], ch.13). For this aid, we use a version of the two-disc scheme presented in the survey [19] and first applied in [21]. Simplifying, one can say that the configurations on the boundary of discs of the first type are longer than the words written on the boundary of corresponding discs of the second type, and the surpluses are relations of the group H . Since every relation of H holds in G , we obtain a homomorphism $H \rightarrow G$ that turns out injective.

To estimate the space function of the group G we use van Kampen diagrams and

induct on the number of hubs. The basic trouble is to cut a diagram Δ having at least two hubs into two subdiagrams with hubs, so that the perimeters of the subdiagrams do not exceed the perimeter of Δ . We have to introduce a new metric where the length of a word depends on syllable factorization of it. To find a short cut we use the mirror symmetry of sectors in discs, but unfortunately for the two-disc scheme, one of the sectors has no (mirror) copies, which creates technical obstacles.² We study both exact and non-accurate copying for various types of complete and incomplete sectors. A number of concepts, e.g. bands, trapezia, discs were incorporated in the algorithmic group theory early, in papers [25], [18], [4] and subsequent ones, and we reproduce them in Section 3 (and partly in sections 2 and 4) in the form they are used now. Some others (replica, unfinished diagram, simple disc) are new.

We mean to consider similar problems for semigroups where the simulating of machines is easier, and for the definition of space function, one does not need cyclic shifts and fragmentation of words.

2 Machines

2.1 Definitions

We will use a model of TM which is close to that in [25]. Recall that a (multi-tape) TM has k tapes and k heads. One can view it as a tuple

$$M = \langle X, Y, Q, \Theta, \vec{s}_1, \vec{s}_0 \rangle$$

where X is the input alphabet, $Y = \sqcup_{i=1}^k Y_i$ is the tape alphabet, $Y_1 \supset X$, $Q = \sqcup_{i=1}^k Q_i$ is the set of states of the heads of the machine, Θ is a set of transitions (commands), \vec{s}_1 is the k -vector of start states, \vec{s}_0 is the k -vector of accept states. (\sqcup denotes the disjoint union.) The sets X, Y, Q, Θ are finite.

We assume that in the normal situation the machine starts working with states of the heads forming the vector \vec{s}_1 , with the head placed at the right end of each tape, and accepts if it reaches the state vector \vec{s}_0 . In general, the machine can be turned on in any configuration and turned off at any time.

A *configuration* of a tape number i of a TM is a word uqv where $q \in Q_i$ is the current state of the head, u is the word to the left of the head, and v is the word to the right of the head. A tape is *empty* if u, v are empty words.

A *configuration* U of a TM is a word

$$\alpha_1 U_1 \omega_1 \alpha_2 U_2 \omega_2 \dots \alpha_k U_k \omega_k$$

where U_i is the configuration of tape i , and α_i, ω_i are special separating symbols. For unification of notation, we shall treat α_i, ω_i as heads of the machine too. These heads correspond to tapes that are always empty and do not change during a computation.

An *input configuration* is a configuration where all tapes except the first one are empty, the configuration of the first tape (let us call it the *input tape*) is of the form uq , $q \in Q_1$, u is a word in the alphabet X , and the states form the start vector \vec{s}_1 . The *accept configuration* is the configuration where the state vector is \vec{s}_0 , the accept vector of the

²Note that one can give shorter proofs of Theorem 1.4 and Corollary 1.6 which do not need the two-disc scheme; but here we just obtain these results after Theorem 1.2 is proved.

machine, and all tapes are empty. (The requirement that the tapes must be empty is often removed for auxiliary machines which are used as parts in constructions of bigger machines.)

To every $\theta \in \Theta$ we correspond a command (marked by the same letter θ), i.e., a pair of sequences of words $[V_1, \dots, V_k]$ and $[V'_1, \dots, V'_k]$ such that for each $j \leq k$, either both V_j and V'_j are configurations of the tape number j or $V_j = \alpha_j q$ and $V'_j = \alpha_j q'$ ($q, q' \in Q_j$), or $V_j = q\omega_j$ and $V'_j = q'\omega_j$ ($q, q' \in Q_j$).

In order to execute this command, the machine checks if V_i is a subword of the configuration of tape i for each $i \leq k$ and if this condition holds the machine replaces V_i by V'_i for all $i = 1, \dots, k$.

Let we have a sequence of configurations w_0, \dots, w_t and a word $h = \theta_1 \dots \theta_t$ in the alphabet Θ , such that for every $i = 1, \dots, t$ the machine passes from w_{i-1} to w_i by applying the command θ_i . Then the sequence $(w_0 \rightarrow w_1 \rightarrow \dots \rightarrow w_t)$ is said to be a *computation with history h* . In this case we shall write $w_0 \circ h = w_t$. The number t will be called the *time* or *length* of the computation.

A configuration w is called *accepted* by a machine M if there exists at least one computation which starts with w and ends with the accept configuration. We do not only consider deterministic TM , for example, we allow several transitions with the same left side. Moreover, for non-deterministic TM , we allow in this paper, to correspond identically equal executions to different symbols $\theta, \theta' \in \Theta$.

A word u in the input alphabet X is said to be *accepted* by the machine if the corresponding input configuration is accepted. The set of all accepted words over the alphabet X is called the *language \mathcal{L}_M recognized by the machine M* .

Let $|w_i|_a$ ($i = 0, \dots, t$) be the number of tape letters (or tape squares) in the configuration w_i . (As in [25], the tape letters are called *a*-letters.) Then the maximum of all $|w_i|_a$ will be called the *space of computation $C : w_0 \rightarrow w_1 \rightarrow \dots \rightarrow w_t$* and will be denoted by $space_M(C)$. By $space_M(w)$, we denote the minimal natural number s such that there is an accepted computation of space at most s , starting with the configuration w . If $u \in L_M$, then, by definition, $space_M(u)$ is the space of the corresponding input configuration w .

The number $S(n) = S_M(n)$ is the minimum of the numbers $space(u)$ over all words $u \in \mathcal{L}_M$, with $||u|| \leq n$. The function $S(n)$ will be called the *space function* of the Turing machine.

The *space of a computation* and the *space function* of M are defined similarly, but one does not count the *a*-letters on the input tape.

The definitions of the *generalized space function* $S'(n) = S'_M(n)$ is also similar to the definition of space function but we consider arbitrary accepted configurations w with $|w|_a = n$, not just the input configurations as in the definitions of $S(n)$. It is clear that $S(n) \leq S'(n)$.

To obtain the definitions of *time_M(w)*, *time_M(u)*, *time function $T_M(n)$* and *generalized time function $T'_M(n)$* , one should replace 'space' for 'time' in the previous definitions.

Given an NTM M , one can add additional states and two more commands so that only input configurations involve the state letters from \vec{s}_1 and only one command applicable to the input configurations, and there is a unique accept configuration \vec{s}_0 with a unique accepting command. In this case we will say that the machine satisfies the *\vec{s}_{10} -condition*. This assumption changes neither the language \mathcal{L} no the functions $S_M(n)$ and $S'_M(n)$.

2.2 Machines with equivalent space and generalized space functions

In this subsection, we construct an NTM M_2 which depends on an NTM M_1 , and prove Lemma 2.1.

Let an NTM M_1 have k tapes, and the first tape of it be the input tape. Then we add a tape number $k + 1$, which is empty for input configurations, and organize the work of the 3-stage machine M_2 as a sequential work of the following machines M_{21} , M_{22} , and M_{23} .

The machine M_{21} uses only one command θ_* that does not change states and adds one square with an auxiliary letter $*$ to the $(k + 1)$ -th tape. M_{21} can execute this command arbitrary many times while the tapes number $1, \dots, k$ leave unchanged the copy of an input configuration of M_1 . Then a connecting rule θ_{12} changes all states of the heads and switches on the machine M_{22} .

The work of M_{22} on the tapes with numbers $1, \dots, k$ copies the work of M_1 , but we extend every command θ of M_1 to the $(k + 1)$ -th tape as θ' so that an application of θ' does not change the current space, that is, if the application of θ inserts m_1 tape squares and deletes m_2 tape squares, then θ' inserts $m_2 - m_1$ (deletes $m_1 - m_2$) squares with letter $*$ on the $(k + 1)$ -th tape if $m_1 - m_2 \leq 0$ (if $m_1 - m_2 \geq 0$). However one cannot apply θ' if $m_1 - m_2$ exceeds the current number of squares on the tape number $k + 1$.

The connecting command θ_{23} is applicable when M_{22} obtains the accept configuration on the first k tapes. It changes the states and switches on the machine M_{23} erasing one by one all squares on the $(k + 1)$ -th tape, and then M_2 accepts. (The tape alphabet of M_{23} has only one letter $*$.)

Let w be a configuration of the machine M_2 such that $w \circ \theta_*$ is defined, or w be obtained after an application of the connecting command θ_{12} . Then we have an input configuration on the tapes with numbers $1, \dots, k$ (plus several $*$ -s on the $(k + 1)$ -th tape). We will denote by $u(w)$ the input word u written on the first tape. It is an input word for the machine M_1 as well, and if it is accepted by M_1 , the expression $space_{M_1}u(w)$ makes sense.

The connecting commands θ_{12} and θ_{23} are not invertible in M_2 by definition. Therefore every non-empty accepting computation of M_2 has history of the form $h_1h_2h_3$, or h_2h_3 , or h_3 , where h_l is the history for M_{2l} , ($l = 1, 2, 3$). (To simplify notation we can attribute the command θ_{12} (the command θ_{23}) to h_2 (to h_3).)

Lemma 2.1. (a) *The machines M_1 and M_2 recognize the same language \mathcal{L} .* (b) *The space function $S_{M_2}(n)$ and the generalized space function $S'_{M_2}(n)$ of M_2 are both equivalent to $S_{M_1}(n)$.* (c) *If w is an accepted configuration M_2 , and the command θ_* is applicable to w , then $space_{M_2}(w) = \max(space_{M_1}(u(w)), |w|_a)$.*

Proof. Assume that $u \in \mathcal{L} = \mathcal{L}_{M_1}$. Then $u \in \mathcal{L}' = \mathcal{L}_{M_2}$ because machine M_{21} can insert sufficiently many squares (equal to the $space_{M_1}(u) - ||u||$) so that the accepting computation of M_1 can be simulated by M_{22} . Also it is clear from the definition of M_2 , that every accepting computation for M_2 having a history $h_1h_2h_3$ as above, simulates, at stage 2, an accepting computation of M_1 with history h_2 . Therefore $\mathcal{L}' = \mathcal{L}$ and $S_{M_1}(n) = S_{M_2}(n)$.

Assume now that $C : w = w_0 \rightarrow \dots \rightarrow w_n$ is an accepting computation of M_2 with $space_{M_2}(C) = space_{M_2}(w)$ and $h = h_1h_2h_3$ is the history with the above factorization (h_1 or h_1h_2 can be empty here). If the word h_1 is empty, then $||w_0|| \geq \dots \geq ||w_n||$ by the definition of the machines M_{22} and M_{23} . Hence the space of this computation is equal to

$|w|_a$ ³. Then let h_1 be non-empty. It follows that the machine M_2 starts working with a copy of an input configuration of the machine M_1 , i.e., the input tape of this configuration contains an input word $u = u(w)$, and the additional $(k+1)$ -th tape has m squares for some $m \geq 0$. Moreover $u \in \mathcal{L}$ since the computation of M_2 is accepting. We consider two cases.

Case 1. Suppose $m \geq \text{space}_{M_1}(u) - \|u\|$. This inequality says that the additional tape has enough squares to enable M_{22} simulating the accepting computation of M_1 with the input word u . Hence there is an M_2 -computation $w_0 \rightarrow \dots \rightarrow w_{n'}$ with history of the form $h'_2 h'_3$, and so its space, as well as the space of our original accepting computation is $|w|_a$.

Case 2. Suppose $m < \text{space}_{M_1}(u) - \|u\|$. Then there is a computation $w_0 \rightarrow \dots \rightarrow w_{n'}$ such that the commands of the M_{21} -stage of it insert squares until the total number of the squares of the $(k+1)$ -th tape becomes equal to $\text{space}_{M_1}(u) - \|u\|$, and then the machines M_{21} and M_{23} work in their standard manner. The space of this (and the original) computation is $\text{space}_{M_1}(u)$.

The estimates obtained in cases 1 and 2 prove the statement (c) of the lemma. They also show that

$$S_{M_1}(n) = S_{M_2}(n) \leq S'_{M_2}(n) \leq \max(S_{M_1}(n), n) \sim S_{M_1}(n)$$

and the statement (b) is completely proved too. \square

2.3 Symmetric machines

For every command θ of a TM , given by the vector $[V_1 \rightarrow V'_1, \dots, V_k \rightarrow V'_k]$, the vector $[V'_1 \rightarrow V_1, \dots, V'_k \rightarrow V_k]$ gives also a command of some TM . These two commands θ and θ^{-1} are called *mutually inverse*.

From now we will assume that the machine M_1 we started in Subsection 2.2 is a DTM and satisfies the \vec{s}_{10} -condition.

Since the machine M_1 is deterministic, we have no invertible commands of the machine M_2 . The definition of the *symmetric* machine $M_3 = M_2^{\text{sym}}$ is the following. Suppose $M_2 = \langle X, Y, Q, \Theta, \vec{s}_1, \vec{s}_0 \rangle$. Then by definition, $M_2^{\text{sym}} = \langle X, Y, Q, \Theta^{\text{sym}}, \vec{s}_1, \vec{s}_0 \rangle$, where Θ^{sym} is the minimal *symmetric* set containing Θ , that is, with every command $[V_1 \rightarrow V'_1, \dots, V_{k+1} \rightarrow V'_{k+1}]$ it contains the inverse command $[V'_1 \rightarrow V_1, \dots, V'_{k+1} \rightarrow V_{k+1}]$; in other words, $\Theta^{\text{sym}} = \Theta^+ \sqcup \Theta^-$, where $\Theta^+ = \Theta$ and $\Theta^- = \{\theta^{-1} \mid \theta \in \Theta\}$.

A computation $w_0 \rightarrow \dots \rightarrow w_t$ of M_3 (or of other machine) is called *reducible* if its history is a reduced word. If the history $h = \theta_1 \dots \theta_t$ contains a subword $\theta_i \theta_{i+1}$, where the commands θ_i and θ_{i+1} are mutually inverse, then obviously there is a shorter computation $w_0 \rightarrow \dots \rightarrow w_{i-1} = w_{i+1} \rightarrow \dots \rightarrow w_t$ whose space does not exceed the space of the original computation.

Lemma 2.2. *Let $w = w_0 \rightarrow w_1 \rightarrow \dots \rightarrow w_t$ be an accepted reduced computation of the machine M_3 , and the command θ_* be applicable to w . Then*

- (a) *the word $u(w)$ belongs to the language \mathcal{L} recognized by M_1 and*
- (b) *the space of this computation is at least $\text{space}_{M_1}(u(w))$.*

³ Here and further we keep in mind that the difference $\|w_i\| - |w_i|_a$ is a constant for any computation.

Proof. Let $h = \theta(1) \dots \theta(t)$ be the history of the computation. If for some i , $w_{i+1} = w_i \circ \theta(i+1)$ where $\theta(i+1) = \theta_{23}$ or $\theta(i+1)^{\pm 1}$ is a command of M_{23} , then one can modify our accepted computation so that, for $j > i+1$, every command $\theta(j)$ is a command of M_{23} and $\|w_j\| \leq \|w_{j-1}\|$. Hence we may assume that h has exactly one letter θ_{23} followed by the commands of M_{23} only, and $h = h_0 \tau_1 h_1 \tau_2 \dots \tau_s h_s$, where $\tau_s = \theta_{23}$, $\tau_i = \theta_{12}^{(-1)^{s-i-1}}$ for $i < s$, and the subwords h_i -s contain no connecting commands. We may also assume that the subword h_0 is empty since the command θ_* does not change the subword $u(w)$.

Since, by the \vec{s}_{10} -condition, only one command of the machine M_1 (and of its analog M_{22}) accepts, the last command of h_{s-1} is this unique command of M_{22} , and so this last command is positive. Therefore if h_{s-1} contains a letter θ^{-1} , where θ is a command of M_{22} , then h has a 2-letter subword $\theta_1^{-1} \theta_2$, where both θ_1 and θ_2 are commands of M_{22} . Hence there is a configuration w_i such that both θ_1 and θ_2 are applicable to w_i . This is impossible since the machine M_1 is deterministic and the history h_{s-1} is a reduced word. Therefore h_{s-1} is entirely the history of a computation of M_{22} , the computation $w_{i_{s-1}} \rightarrow \dots \rightarrow w_t$ with history $\tau_{s-1} h_{s-1} \tau_s h_s = \theta_{12} h_{s-1} \theta_{23} h_s$ is an accepted computation of M_2 , the word $u(w_{i_{s-1}})$ belongs to the language \mathcal{L} , and the space of the computation $w_{i_{s-1}} \rightarrow \dots \rightarrow w_t$ is at least $\text{space}_{M_1}(u(w_{i_{s-1}}))$ by Lemma 2.1 (c).

Assume, by induction on j , that $u(w_{i_{s-j}})$ belongs to \mathcal{L} for $j \geq 1$, where the computation $w_{i_{s-j}} \rightarrow \dots \rightarrow w_t$ has history $\tau_{s-j} h_{s-j} \dots \tau_s h_s$, and the space of this computation is at least $\text{space}_{M_1}(u(w_{i_{s-j}}))$.

Then the word $w_{i_{s-j-1}}$ has similar properties if h_{s-j-1} consists of the commands of M_{21} or their inverses since these commands do not change the content of the tapes number $1, \dots, k$. Otherwise the commands of h_{s-j-1} are commands of M_{22} (and inverses), and since this machine is deterministic, the word h_{s-j-1} has no subwords $\theta_1^{-1} \theta_2$ with positive θ_1 and θ_2 . Therefore we have $h_{s-j-1} = g' g''^{-1}$, where both g' and g'' are (positive) histories of M_{22} -computations. This implies the equality $(w_{i_{s-j-1}} \circ \tau_{s-j-1}) \circ g' = w_{i_{s-j}} \circ g''$. Since the commands $\theta_{12}^{\pm 1}$ do not change $u(w_i)$ -s, we have $W_{i_{s-j-1}} \circ g' = W_{i_{s-j}} \circ g''$, where $W_{i_{s-j-1}}$ and $W_{i_{s-j}}$ are the input configurations for the machine M_1 with inputs $u(w_{i_{s-j-1}})$ and $u(w_{i_{s-j}})$, respectively. (Here we use identical letters for the corresponding commands of M_1 and M_{22} .)

The machine M_1 is deterministic, and so the accepted computation for $W_{i_{s-j}}$ must look like $W_{i_{s-j}} \rightarrow \dots \rightarrow W_{i_{s-j}} \circ g'' \rightarrow \dots$, and consequently, the configuration $W_{i_{s-j}} \circ g''$ is accepted by M_1 . Therefore we can construct the accepted computation $W_{i_{s-j-1}} \rightarrow \dots \rightarrow W_{i_{s-j-1}} \circ g' = W_{i_{s-j}} \circ g'' \rightarrow \dots$ for M_1 , and so the word $u(w_{i_{s-j-1}})$ belongs to \mathcal{L} , as desired.

The constructed accepted computation of M_1 is decomposed in two parts. It follows from the definition of M_{22} that the space of the first part is majorized by the space of the M_3 -computation $w_{i_{s-j-1}} \rightarrow \dots \rightarrow w_{i_{s-j-1}} \circ g'$ which is a part of the computation $w_{i_{s-j-1}} \rightarrow \dots \rightarrow w_t$. The second part is a part of the deterministic accepted M_1 -computation with input $u(w_{i_{s-j}})$, and so, by the inductive hypothesis, the space of this part does not exceed the space of the M_3 -computation $w_{i_{s-j}} \rightarrow \dots \rightarrow w_t$. Hence $\text{space}_{M_1}(u(w_{i_{s-j-1}}))$ does not exceed the space of the M_3 -computation $w_{i_{s-j-1}} \rightarrow \dots \rightarrow w_t$.

Since $w = w_{i_1}$, the lemma is proved by induction on j . \square

Lemma 2.3. *The machines M_1 and M_3 recognize the same language. The generalized space functions $S'_{M_2}(n)$ and $S'_{M_3}(n)$ are equivalent.*

Proof. We recall that every computation of the machine M_2 is also a computation of M_3 .

Therefore the first statement follows from lemmas 2.1 (a) and 2.2 (a).

To prove the second part, it suffices to prove that for every accepted configuration w of M_2 (of M_3), there is an accepted configuration w' of M_3 (of M_2) such that $\|w'\| \leq \|w\|$ but $\text{space}_{M_3}(w') \geq \text{space}_{M_2}(w)$ (respectively, $\text{space}_{M_2}(w') \geq \text{space}_{M_3}(w)$).

(1) Consider an accepted computation $w = w_0 \rightarrow \dots \rightarrow w_t$ of M_2 whose space is equal to $\text{space}_{M_2}(w)$. If the first command of this computation is not a command of M_{21} , then $\|w_0\| \geq \|w_1\| \geq \dots \geq \|w_t\|$, and therefore $\text{space}_{M_2}(w) = |w|_a \leq \text{space}_{M_3}(w)$, and so one can choose $w' = w$. If the first command is a command of M_{21} , then by lemmas 2.1(c) and 2.2 (b), we have $\text{space}_{M_2}(w) = \max(\text{space}_{M_1}(u(w)), |w|_a) \leq \text{space}_{M_3}(w)$, and again $w' = w$.

(2) Now consider a reduced accepted computation $w = w_0 \rightarrow \dots \rightarrow w_t$ of M_3 whose space is equal to $\text{space}_{M_3}(w)$. If the first command (or the inverse of it) is a command of M_{23} , then the commands of the shortest accepted computation with minimal space just erase squares. Hence $\text{space}_{M_3}(w) = |w|_a = \text{space}_{M_2}(w)$, and we can choose w' equal to w . If the first command is a command of M_{21}^{sym} , then by Lemma 2.2 (a), the word w is accepted by M_2 . Since every command of M_2 is a command of M_3 , we have $\text{space}_{M_3}(w) \leq \text{space}_{M_2}(w)$, and again it suffices to set $w' = w$.

Thus, we may assume that the first command of our computation (or the inverse) is a command of M_{22} . Therefore the history h of this computation has a prefix $h'h''$ with non-empty h'' , where every command of h' is command of M_{22}^{sym} and either every command of h'' (or the inverse command) is a command of M_{21} or every command of h'' is a command of M_{23} . In the latter case, we may assume that $h = h'h''$ and $\|w_0\| \geq \|w_1\| \geq \dots \geq \|w_t\|$, and so $\text{space}_{M_3}(w) = |w|_a \leq \text{space}_{M_2}(w)$, and $w' = w$. In the former case, we set $w' = w \circ h'$ and note that $\|w\| = \|w_1\| = \dots = \|w'\|$ since the commands of the computation $w \rightarrow \dots \rightarrow w'$ with history h' do not change the number of tape squares. In particular, we have $\text{space}_{M_3}(w) \leq \text{space}_{M_3}(w')$. Since the command θ_* is applicable to w' in this case, we have $\text{space}_{M_3}(w') \leq \text{space}_{M_2}(w')$ as this was observed in the previous paragraph. Therefore $\text{space}_{M_3}(w) \leq \text{space}_{M_3}(w') \leq \text{space}_{M_2}(w')$, as desired; and the lemma is proved. \square

Lemma 2.4. *For every DTM M recognizing a language \mathcal{L} and having a space function $S(n)$ (for every NTM M recognizing a language \mathcal{L} and having an FSC space function $f(n)$), there exists an NTM M' with the following properties.*

1. *The machine M' recognizes the language \mathcal{L} .*
2. *M' is symmetric.*
3. *The space and the generalized space functions of M' are equivalent to $S(n)$ (respectively, are equivalent to $f(n)^2$).*
4. *For every command $[V_1 \rightarrow V'_1, \dots, V_k \rightarrow V'_k]$ of M' , we have $\sum |V_i|_a + \sum |V'_i|_a \leq 1$, i.e., at most one tape letter is involved in the command.*

Proof. Assume that the machine M is deterministic. Then starting with $M_1 = M$, we construct machine M_2 described in Subsection 2.2 and machine $M_3 = M_2^{\text{sym}}$. Now Lemma 2.3 implies statement 1, and statement 2 is true since $M_3 = M_2^{\text{sym}}$. Then, by lemmas 2.3 and 2.1 (b), we have

$$S'_{M_3}(n) \sim S'_{M_2}(n) \sim S_{M_2}(n) \sim S_{M_1}(n) = S(n)$$

, and $S(n) \leq S_{M_3}(n) \leq S'_{M_3}(n)$ by Lemma 2.2 (b), and so all these functions are equivalent. Finally, we modify M_3 to obtain property 4. For example, if for a one-tape machine we have a command $aq \rightarrow bq'$, then we introduce a new state letter q'' and replace this command by two commands $aq \rightarrow q''$ and $q'' \rightarrow bq'$. It is easy to see that the obtained machine M' satisfies property 4 and keeps holding properties 1 – 3 of M_3 .

If M is non-deterministic, then we first use that the function $f(n)$ is *FSC*, and therefore, by Savitch's theorem ([9], Theorem 1.30), there exists a *DTM* M_1 accepting the same language L with space function equivalent to $f(n)^2$. So the replacement of $S(n)$ by $f(n)^2$ in the previous paragraph provides the proof of the non-deterministic version of the lemma. \square

2.4 S-machines

Ordinary Turing machines work with positive words and they can see letters on the tape near the position where the head is. The command executed by the machine depends not only on the state of the head but also on the letter(s) observed by the head. In contrast, S-machines introduced in [SBR] work with words in group alphabets and they are almost "blind", i.e., the heads do not observe the tape letters. But the heads can "see" each other if there are no tape letters between them. We will use the following precise definition of an *S-machine* S .

Let k be a natural number. Consider a *language of admissible words*. It consists of words of the form

$$q_1 u_1 q_2 \dots u_k q_{k+1},$$

where q_i are letters from disjoint sets Q_i ($i = 1, \dots, k+1$), u_i are reduced words in a group alphabet Y_i , (i.e. every letter a belongs to it together with the inverse letter a^{-1}) and the sets $Y = \sqcup Y_i$ and $Q = \sqcup Q_i$ are finite. The letters from Q are called *state* letters, the letters from Y are *tape* letters. Notice that in every admissible word, there is exactly one representative of each Q_i and these representatives appear in this word in the order of the indexes of Q_i . (i.e., unlike [22], we consider only the regular order of Q_i -s in admissible words).

There is a finite set of *commands* (or *rules*) Θ . To every $\theta \in \Theta$, we associate two sequences of reduced words from the free group $F(Q \cup Y)$: $B(\theta) = [U_1, \dots, U_{k+1}]$, $T(\theta) = [V_1, \dots, V_{k+1}]$, and a subset $Y(\theta) = \sqcup Y_i(\theta)$ of Y , where $Y_i(\theta) \subseteq Y_i$.

The words U_i, V_i satisfy the following restriction:

(*) For every $i = 1, \dots, k+1$, the words U_i and V_i have the form

$$U_i = v_{i-1} q_i u_i, \quad V_i = v'_{i-1} q'_i u'_i$$

where $q_i, q'_i \in Q_i$, u_i and u'_i are words in the alphabet $Y_i(\theta)$, v_{i-1} and v'_{i-1} are words in the alphabet $Y_{i-1}(\theta)$. The words $v_0, v'_0, u_{k+1}, u'_{k+1}$ are empty.

Sometimes we will denote the rule θ by $[U_1 \rightarrow V_1, \dots, U_{k+1} \rightarrow V_{k+1}]$. This notation contains no information about the sets $Y_i(\theta)$. In most cases it will be clear what these sets are. In the *S-machines* used in this paper, the sets $Y_i(\theta)$ will be mostly equal to either Y_i or \emptyset . By default $Y_i(\theta) = Y_i$.

In order to simplify the notation, we will use the notation $v_i q_i u_i \xrightarrow{\ell} v'_i q'_i u'_i$ for a part of a rule when the corresponding $Y_i(\theta)$ is empty (a similar notation has been used in [20]).

Every S -rule $\theta = [U_1 \rightarrow V_1, \dots, U_{k+1} \rightarrow V_{k+1}]$ has an inverse $\theta^{-1} = [V_1 \rightarrow U_1, \dots, V_{k+1} \rightarrow U_{k+1}]$; we set $Y_i(\theta^{-1}) = Y_i(\theta)$. We always divide the set of rules Θ of an S -machine into two disjoint parts, Θ^+ and Θ^- such that for every $\theta \in \Theta^+$, $\theta^{-1} \in \Theta^-$ and for every $\theta \in \Theta^-$, $\theta^{-1} \in \Theta^+$. The rules from Θ^+ (resp. Θ^-) are called *positive* (resp. *negative*).

An S -machine is a rewriting system. To apply an S -rule θ to an admissible word $W = q_1 w_1 q_2 \dots w_k q_{k+1}$ means to check if every w_i is a word in the alphabet $Y_i(\theta)$ and then, if W satisfies this condition, to replace simultaneously subwords U_i by subwords V_i ($i = 1, \dots, k+1$). This replacement is allowed to perform in the form $q_i \rightarrow v'_{i-1} v_i^{-1} q'_i u_i^{-1} u'_i$ followed by the reducing of the resulted word. The following convention is important in the definition of S -machine: *After every application of a rewriting rule, the word is automatically reduced. The reducing is not considered a separate step of an S -machine.*

The definitions of computation, its history, input admissible words, the accept word, the language of admissible words, space of a computation, space and generalized space functions, time and generalized time functions of an S -machine are similar to those for TM . (One should replace the word "configuration" by "admissible word" in the definitions.)

Although S -machines are usually highly non-deterministic, they better adapted to simulating by finitely presented groups than ordinary TM (and moreover, S -machines are treated in [22] as HNN-extensions of free group with basis $Y \cup Q$). On the other hand, it is mentioned in [25] that every symmetric NTM M can be viewed as an S -machine $S(M)$: just interpret the commands of the Turing machine as S -rules. (For example, the part of a rule of the form $aq \rightarrow bq'$ is interpreted as $q \rightarrow a^{-1}bq'$, and the part of the form $\alpha_j q \rightarrow \alpha_j q'$ is interpreted by the pair $\alpha_j \xrightarrow{\ell} \alpha_j$, $q \rightarrow q'$.) Unfortunately, the language recognized by $S(M)$ is in general much bigger than the language recognized by M since M works with a *positive* tape alphabet only. Nevertheless the following statement is true:

Lemma 2.5. *(Compare with Prop. 4.1[25].) Every computation of a symmetric NTM M is a computation of $S(M)$ with the same history. If M satisfies property 4 from Lemma 2.4, then every positive computation of $S(M)$, i.e., a computation consisting of positive words, is a computation of M with the same history.*

Proof. Every positive admissible word W of $S(M)$ is a configuration of the Turing machine M . Assume that a rule $\bar{\theta}$ of $S(M)$ corresponding to a command θ of M is applicable to this W and the word $W \circ \bar{\theta}$ is positive. Recall that by property 4 of Lemma 2.4, θ involves at most one tape letter (e.g., it cannot replace a tape letter by a tape letter or have a part of the form $aq \rightarrow aq'$). Therefore the positiveness of both W and $W \circ \bar{\theta}$ implies that the application of $\bar{\theta}$ just coincides with the application of θ . The statement of the lemma follows. \square

2.5 Composition with an adding machine

Further we use the auxiliary adding S -machine $Z(A)$ from [22]. In [22], the main duty of $Z(A)$ was the exponential slowing down of basic computations, while now we will mainly use the capacity of $Z(A)$ (observed in Lemma 3.25 (2) [22]) to check whether an admissible word is positive or not.

The tape alphabet of $Z(A)$ consists of an alphabet $A^{\pm 1}$ and two copies $A_0^{\pm 1}$ and $A_1^{\pm 1}$ of $A^{\pm 1}$ while the input alphabet is $A_0^{\pm 1}$. The admissible words are of the form $LupvR$, where

u is a reduced word in the alphabet $A_0^{\pm 1} \cup A_1^{\pm 1}$, v is a reduced word in $A^{\pm 1}$, the symbols L, p, R , are state letters, the commands do not change L and R , and $p \in \{p(1), p(2), p(3)\}$. The input configurations have form $Lup(1)R$ and the accept ones are of the form $Lup(3)R$. The list of rules is given in subsection 3.6 of [22], but we will not use them here and rather formulate, in Lemma 2.6, the required properties obtained in [22].

If w a word in the alphabet $A_0^{\pm 1} \cup A_1^{\pm 1}$, then its *projection* onto $A^{\pm 1}$ takes every letter to its copy in A .

Lemma 2.6. *The following properties of the machine $Z(A)$ hold.*

(1) *Every positive input word u in the alphabet A_0 is accepted by a canonical computation of $Z(A)$ with a positive history and equal lengths of all words appearing in this computation.*

(2) *For every computation $LupvR = w \rightarrow \dots \rightarrow w' = Lu'p'v'R$ of $Z(A)$, the projections of the words uv and $u'v'$ onto A are freely equal. In particular, $u = u'$ if the words v and v' are empty and the words u and u' contain no letters from $A_1^{\pm 1}$.*

(3) *If $w_0 \rightarrow \dots \rightarrow w_t$ is a reduced computation of $Z(A)$ and $\|w_0\| < \|w_1\|$, then $\|w_1\| \leq \|w_2\| \leq \dots \leq \|w_t\|$.*

(4) *For every reduced computation $w_0 \rightarrow \dots \rightarrow w_t$, we have $\|w_i\| \leq \max(\|w_0\|, \|w_t\|)$ ($i = 0, \dots, t$).*

(5) *If $w = LupR$, where $p = p(1)$ (or $p = p(3)$), $w = w_0 \rightarrow \dots \rightarrow w_t$ is a reduced computation, w_t contains the subword $p(3)R$ (respectively, $p(1)R$), and all a -letters of w_0 and w_t are from $A_0^{\pm 1}$, then u is a positive word and all words of the computation have the same length. The length of this computation is at least $2^{\|u\|}$.*

(6) *There no reduced computation $w = w_0 \rightarrow \dots \rightarrow w_t$ of length $t \geq 1$ such that both w_0 and w_t contain $p(1)R$ or both of them contain $p(3)R$ and all a -letters of w_0 and w_t belong to $A_0^{\pm 1}$.*

Proof. (1) This computation is described in [22], p. 1344.

(2) This claim is the statement of Lemma 3.18 of [22].

(3) This statement is true by Lemma 3.24 of [22]

(4), (5) These statements are contained in Lemma 3.25 of [22].

(6) This statement is contained in Lemma 3.27 of [22].

□

We somewhat modify the definition of the composition of a symmetric Turing machine M and the adding machine $Z(A)$, given in [22]. The difference is that machines of the form $Z(A)$ will work not only after the application of every command of M but before applications of commands from M as well. This make possible to simulate the work of any symmetric NTM, not only S -machine as that was done in [22]. So the aim of following interbreeding is to obtain an S -machine \mathcal{S} which recognize the same language and has the same space and generalized space function as the symmetric Turing machine M . The S -machine constructed in [SBR] cannot serve in the present paper since the space and the generalized space functions of that machine are equivalent to the time function.

Consider a symmetric NTM $M = \langle X, Y, Q, \Theta, \vec{s}_1, \vec{s}_0 \rangle$ with $Y = \sqcup_{i=1}^l Y_i$, and with the \vec{s}_{10} -condition. The set Θ is a disjoint union of positive and negative commands: $\Theta = \Theta^+ \sqcup \Theta^-$. Let $S(M)$ be the associated S -machine defined before Lemma 2.5. We will assume that the admissible words of $S(M)$ are of the form $k_1 u_1 k_2 \dots u_l k_{l+1}$, where k_i are letters from disjoint sets Q_i ($i = 1, \dots, l+1$), u_i is a reduced words in the alphabet Y_i ($i \leq l$).

To define the composition $\mathcal{S} = M \circ Z$ of M and $Z(A)$, we will insert a p -letter between any two consecutive q -letters k_i, k_{i+1} in an admissible word of $S(M)$, to be able to treat any subword $k_i \dots p \dots k_{i+1}$ as an admissible word for a copy of $Z(A)$. In particular, the *start* and the *accept* words of \mathcal{S} are obtained from, respectively, the start and accept words of M .

First, for every $i = 1, \dots, l$, we make two copies of the alphabet Y_i of $S(M)$ ($i = 1, \dots, l$): $Y_{i,0} = Y_i$ and $Y_{i,1}$. The set of state letters of the new machine is

$$K_1 \sqcup P_1 \sqcup K_2 \sqcup P_2 \sqcup \dots \sqcup P_l \sqcup K_{l+1},$$

where $P_i = \{p_i, p_i(\theta, 1^-), p_i(\theta, 2^-), p_i(\theta, 3^-), p_i(\theta, 1^+), p_i(\theta, 2^+), p_i(\theta, 3^+) \mid \theta \in \Theta^+\}$, $i = 1, \dots, l$.

The set of state letters is

$$\bar{Y} = (Y_{1,0} \sqcup Y_{1,1}) \sqcup Y_1 \sqcup (Y_{2,0} \sqcup Y_{2,1}) \sqcup Y_2 \sqcup \dots \sqcup (Y_{l,0} \sqcup Y_{l,1}) \sqcup Y_l;$$

the components of this union will be denoted by $\bar{Y}_1, \dots, \bar{Y}_{2l}$.

The set of positive rules $\bar{\Theta}$ of $M \circ Z$ is a union of the set of modified positive rules of $S(M)$ and of positive rules of $Z_i(\theta, -)^+$ and $Z_i(\theta, +)^+$ ($\theta \in \Theta, i = 1, \dots, l$) which are copies of the machines $Z(Y_i)$ (also suitably modified).

More precisely, let a positive command θ of $S(M)$ differs from the unique start and accept commands of $S(M)$ and have the form

$$[k_1 u_1 \rightarrow k'_1 u'_1, v_1 k_2 u_2 \rightarrow v'_1 k'_2 u'_2, \dots, v_l k_{l+1} \rightarrow v'_l k'_{l+1}]$$

where $k_i, k'_i \in K_i$, u_i and v_i are words in Y_i . Then its copy in $M \circ Z$ is

$$\bar{\theta} = [k_1 u_1 \rightarrow k'_1 u'_1, v_1 p_1(\theta, 3^-) \xrightarrow{\ell} v'_1 p_1(\theta, 1^+), k_2 u_2 \rightarrow k'_2 u'_2, \dots, v_l p_l(\theta, 3^-) \xrightarrow{\ell} v'_l p_l(\theta, 1^+), k_{l+1} \rightarrow k'_{l+1}]$$

with $\bar{Y}_{2i-1}(\bar{\theta}) = Y_{i,0}(\theta)$ and $\bar{Y}_{2i}(\bar{\theta}) = \emptyset$ for every i , in particular, the words u_i, v_i are rewritten here in alphabet $Y_{i,0}$.

Thus the modified rule from $\bar{\theta} \in \bar{\Theta}$ turns on l copies of the machine $Z(A)$ (for different A 's).

Each machine $Z_i(\theta, -)$ is a copy of the machine $Z(Y_i)$, where every rule $\tau = [U_1 \rightarrow V_1, U_2 \rightarrow V_2, U_3 \rightarrow V_3]$ is replaced by the rule of the form

$$\bar{\tau}_i(\theta, -) = \left[\begin{array}{l} \bar{U}_1 \rightarrow \bar{V}_1, \bar{U}_2 \rightarrow \bar{V}_2, \bar{U}_3 \rightarrow \bar{V}_3, \\ k_j \rightarrow k'_j, p_j(\theta, 3^-) \xrightarrow{\ell} p_j(\theta, 3^-), j = 1, \dots, i-1, \\ p_s(\theta, 1^-) \xrightarrow{\ell} p_s(\theta, 1^-), k_{s+1} \rightarrow k_{s+1}, s = i+1, \dots, l \end{array} \right]$$

where $\bar{U}_1, \bar{U}_2, \bar{U}_3, \bar{V}_1, \bar{V}_2, \bar{V}_3$ are obtained from $U_1, U_2, U_3, V_1, V_2, V_3$, respectively, by replacing $p(j)$ with $p_i(\theta, j^-)$, L with k_i and R with k_{i+1} , and for $s \neq i$, $\bar{Y}_{2s-1}(\bar{\tau}_i(\theta, -)) = Y_{i,0}$.

Similarly, each machine $Z_i(\theta, +)$ is a copy of the machine $Z(Y_i)$, where every rule $\tau = [U_1 \rightarrow V_1, U_2 \rightarrow V_2, U_3 \rightarrow V_3]$ is replaced by the rule of the form

$$\bar{\tau}_i(\theta, +) = \left[\begin{array}{l} \bar{U}_1 \rightarrow \bar{V}_1, \bar{U}_2 \rightarrow \bar{V}_2, \bar{U}_3 \rightarrow \bar{V}_3, \\ k'_j \rightarrow k'_j, p_j(\theta, 3^+) \xrightarrow{\ell} p_j(\theta, 3^+), j = 1, \dots, i-1, \\ p_s(\theta, 1^+) \xrightarrow{\ell} p_s(\theta, 1^+), k'_{s+1} \rightarrow k'_{s+1}, s = i+1, \dots, l \end{array} \right]$$

where $\bar{U}_1, \bar{U}_2, \bar{U}_3, \bar{V}_1, \bar{V}_2, \bar{V}_3$ are obtained from $U_1, U_2, U_3, V_1, V_2, V_3$, respectively, by replacing $p(j)$ with $p_i(\theta, j^+)$, L with k'_i and R with k'_{i+1} , and for $s \neq i$, $\bar{Y}_{2s-1}(\bar{\tau}_i(\theta, +)) = Y_{i,0}$.

If θ is the start (the accept) command of $S(M)$, then we introduce only machines $Z_i(\theta, +)$ (only $Z_i(\theta, -)$, respectively), and replace the letters $p_j(\theta, 3^-)$ (replace $p_j(\theta, 1^+)$, resp.) by p_j in the above definition of the command $\bar{\theta}$.

In addition, we need the following *transition* rules $\zeta(\theta, -)$ and $\zeta(\theta, +)$ that transform all p -letters from and to their original.

$$[k_i \rightarrow k_i, p_j \xrightarrow{\ell} p_j(\theta, 1^-), i = 1, \dots, l+1, j = 1, \dots, l].$$

$$[k'_i \rightarrow k'_i, p_j(\theta, 3^+) \xrightarrow{\ell} p_j, i = 1, \dots, l+1, j = 1, \dots, l].$$

If θ is the start (the accept) command of $S(M)$, then we introduce only $\zeta(\theta, +)$ (only $\zeta(\theta, -)$, respectively).

Thus while the machine $Z_i(\theta, -)$ (the machine $Z_i(\theta, +)$) works all other machines $Z_j(\theta, -)$ (all machines $Z_j(\theta, +)$, $j \neq i$) must stay idle (their state letters do not change and do not move away from the corresponding k -letters). After the machine $Z_i(\theta, -)$ (the machine $Z_i(\theta, +)$) finishes, i.e., the state letter $p_i(\theta, 3^\pm)$ appears next to k_{i+1} (next to k'_{i+1}), the next machine $Z_{i+1}(\theta, -)$ (the machine $Z_{i+1}(\theta, +)$) starts working. The transition rule $\zeta(\theta, -)$ switches on the consecutive works of the machines $Z_1(\theta, -), \dots, Z_l(\theta, -)$. After all p -letters have the form $p_j(\theta, 3^-)$ we can apply the rule $\bar{\theta}$ and turn all $p_j(\theta, 3^-)$ into $p_j(\theta, 1^+)$. This switches on the consecutive work of $Z_1(\theta, +), \dots, Z_l(\theta, +)$, followed by the transition rule $\zeta(\theta, +)$.

Thus, in order to simulate a computation of the symmetric *TM* M (and of the *S*-machine $S(M)$) consisting of a sequence of applications of rules $\theta_1, \theta_2, \dots, \theta_s$, we first apply all rules corresponding to θ_1 , then all rules corresponding to θ_2 , then all rules corresponding to θ_3 , etc. The language \mathcal{L}_S of \mathcal{S} consists of some words u in the alphabet $Y_{1,0}$. In particular, every input admissible words of \mathcal{S} is of the form $\Sigma(u) = k_1 u p_1 k_2 p_2 k_3 \dots k_{l-1} p_l k_l$. We denote by $\Sigma_0 = \Sigma_0(\mathcal{S})$ the accept word of \mathcal{S} .

The modified rules $\bar{\theta}$ of $S(M)$ will be called *basic rules* of the *S*-machine \mathcal{S} .

2.6 Computations of machine \mathcal{S}

Given a computation C of \mathcal{S} , one obtains a computation $C_{S(M)}$ of the $S(M)$ after omitting all the (copies of) commands of Z , deleting the additional state letters of Z from the admissible word of the computation C , and replacing the letters from alphabets $Y_{i,0}$ by their copies in Y_i (By Lemma 2.6 (2), the computation $C_{S(M)}$ is well-defined, although C_{S_M} is of length 0 if C has no basic rules.)

Lemma 2.7. (1) *The start and the accept rules of \mathcal{S} are basic rules which are the copies of the start and the accept commands of M , respectively. The machine $M \circ Z$ satisfies the \bar{s}_{10} -condition.*

(2) *If the history h of a reduced computation $C : w_0 \rightarrow \dots \rightarrow w_t$ of \mathcal{S} contains no basic commands, then $\|w_i\| \leq \max(\|w_0\|, \|w_t\|)$ for every i ($0 \leq i \leq t$). If the only basic letter of h is the last one, then $\|w_i\| \leq \|w_0\|$ for $i < t$.*

(3) *If a computation C of \mathcal{S} is reduced then $C_{S(M)}$ is also reduced.*

(4) *For any reduced computation $C : w_0 \rightarrow \dots \rightarrow w_t$ of \mathcal{S} starting and ending with basic rules we have $\text{space}_{\mathcal{S}}(C) = \text{space}_{S(M)}(C_{S(M)})$.*

(5) For every positive reduced computation $C : w_0 \rightarrow \dots \rightarrow w_t$ of the machine $S(M)$, there is a canonical reduced computation C_S of \mathcal{S} whose history starts and ends with basic commands, such that $(C_S)_{S(M)} = C$. Moreover, we have $\text{space}_{\mathcal{S}}(C_S) = \text{space}_{S(M)}(C)$.

Proof. (1) Property (1) follows from the similar property of the machine M and from the definition of the machine $M \circ Z$

(2) We start with the first statement. If the history of the whole computation consists of the commands of one machine $Z_i(\theta, \pm)$, then the statement follows from Lemma 2.6(3). Otherwise it has $s \geq 1$ admissible words w_{i_1}, \dots, w_{i_s} , such that the history of every subcomputation

$$C_0 : w_0 \rightarrow \dots \rightarrow w_{i_1}, \dots, C_j : w_{i_j} \rightarrow \dots \rightarrow w_{i_{j+1}}, \dots, C_s : w_{i_s} \rightarrow \dots \rightarrow w_t$$

either consists of the commands of some $Z_i(\theta, \pm)$, being a maximal subcomputation with this property, or has only one transition letter $\zeta(\theta, \pm)^{\pm 1}$ for some θ . In the former case, all the admissible words participating in C_j have the same length for $j \in [1, s-1]$ by Lemma 2.6 (5,6). The same is clearly true in the latter case. Therefore it suffices to prove that the lengths of the admissible words do not decrease in the subcomputations C_0^{-1} and C_s .

Let us consider C_s only, assuming that it is a computation of some machine $Z_i(\theta, \pm)$. It corresponds to a computation $LupvR = W_0 \rightarrow \dots \rightarrow W_m = Lu'pv'R$ of a machine of the form $Z(A)$ with $m = t - i_s$. Since $s \geq 1$ and C_s is a maximal subcomputation corresponding to $Z_i(\theta, \pm)$, we must have $\|v\| = 0$. (Otherwise only commands of $Z_i(\theta, \pm)$ could be applied to w_{i_s} , and so $w_{i_s-1} \rightarrow w_{i_s} \rightarrow \dots \rightarrow w_t$ were a longer computation of the same $Z_i(\theta, \pm)$.) Hence the projection of the word uv onto A is reducible, and so $\|W_0\| \leq \|W_i\|$ for every $i \in [1, m]$ by Lemma 2.6 (2). Therefore either all W_i -s have equal lengths or $\|W_0\| \leq \|W_1\| \leq \dots \leq \|W_m\|$ by Lemma 2.6 (3). In any case, we have $\|W_0\| \leq \dots \leq \|W_m\|$. This implies $\|w_{i_s}\| \leq \dots \leq \|w_t\|$, as required.

The proof of the second claim is similar: The computation $w_{i_1} \rightarrow \dots \rightarrow w_{i_s}$ is product of subcomputations C_j -s which preserve the lengths of w_i -s, while C_0 and C_s cannot increase the space.

(3) Assume that $\tau_1 \dots \tau_t$ is a history of a computation $w_0 \rightarrow \dots \rightarrow w_t$ of \mathcal{S} , where τ_1 corresponds to a positive command θ of $S(M)$, τ_t corresponds to θ^{-1} , and other rules are not basic. Then non-empty history of the computation $w_1 \rightarrow \dots \rightarrow w_{t-1}$ is a product $H_1 \dots H_s$ where H_i ($i = 1, \dots, s$) are maximal subwords corresponding to some $Z_{j(i)}(\theta_i, \pm)$. Since H_1 and H_s correspond to $Z_1(\theta, +)$, either $s = 1$ or there is i such that both H_{i-1} and H_{i+1} correspond to the same $Z_{j(i) \pm 1}(\theta', \pm)$. It follows that the computation with history H_i satisfies the assumption of Lemma 2.6 (6), a contradiction.

(4) Property (4) follows from (2) and the definition of the computation $C_{S(M)}$.

(5) Given C , the computation C_S with the same space and with property $(C_S)_{S(M)} = C$ is briefly described above at the end of subsection 2.5 and with more details (though with submachines $Z_i(\theta, +)$ but without $Z_i(\theta, -)$) in subsection 3.7 of [22]. \square

Lemma 2.8. Let $C : w_0 \rightarrow \dots \rightarrow w_t$ be a reduced computation of \mathcal{S} such that the first command θ_1 and the last commands $\bar{\theta}_t$ are basic ones, and $C_{S(M)} = W_0 \rightarrow \dots \rightarrow W_s$ with $s \geq 2$. Then the subcomputation $W_1 \rightarrow \dots \rightarrow W_{s-1}$ is positive and $t - 2 \geq 2^{|W_1|_a/l}$. If $\bar{\theta}_1$ (if $\bar{\theta}_t$) is a start (is an accept) command, then the word W_0 (respectively, W_t) is also positive.

Proof. To justify the first claim of the lemma, it suffices to prove that the word W_1 is positive under assumption that there are no basic commands in the computation $w_1 \rightarrow \dots \rightarrow w_{t-1}$, and the word W_1 corresponds to w_1 . Therefore it suffices to prove that the word w_1 is positive.

We first assume that the first command $\bar{\theta}$ of the history of C is a positive basic command. Then $\bar{\theta}$ switches on the machine $Z_1(\theta, +)$. Since $s \geq 2$, this machine must complete its work before the computation C ends. The computation of $Z_1(\theta, +)$ cannot be empty since otherwise $\bar{\theta}$ were followed by $\bar{\theta}^{-1}$ because w_1 involves the state letter $p_1(\theta, 1^+)$. But this would contradict to the reducibility of C .

Hence, by Lemma 2.6 (6), the work of $Z_1(\theta, +)$ soon or later leads to an admissible word w_i containing the state letter $p_1(\theta, 3^+)$. By Lemma 2.6(5), the subword of w_1 of the form $k_1 u_1 p_1(\theta, 1^+) k_2$ is positive, and the time of the work of $Z_1(\theta, +)$ is at least $2^{\|u_1\|}$. Then the machine $Z_1(\theta, +)$ ends working and switches on the machine $Z_2(\theta, +)$, whose work similarly provides the positiveness of the subword of w_1 having form $k_2 u_2 p_2(\theta, 1^+) k_3$. Finally we obtain that w_1 is covered by positive subwords, and so it is positive itself. Besides the time of work of all $Z_1(\theta, +), \dots, Z_l(\theta, +)$ is at least $2^{|W_1|_a/l}$ since $|W_1|_a = \sum_{j=1}^l \|u_j\|$.

If the command $\bar{\theta}^{-1}$ is positive, then it switches on the machine $Z_l(\theta, -)$, and we first obtain the positiveness of $k_l u_l p_l(\theta, 3^-) k_{l+1}$, then the positiveness of $k_{l-1} u_{l-1} p_{l-1}(\theta, 3^-) k_l$, and so on.

Since start and accept commands leave tape letters unchanged, the second statement follows from the positiveness of the word W_1 (of the word W_{t-1}). The lemma is proved. \square

Lemma 2.9. (1) *The S-machine \mathcal{S} and the symmetric NTM M recognize the same language \mathcal{L} .*

(2) *The space functions $S_M(n)$ and $S_{\mathcal{S}}(n)$ of M and \mathcal{S} , respectively, are equivalent.*

(3) *The generalized space functions $S'_M(n)$ and $S'_{\mathcal{S}}(n)$ of M and \mathcal{S} are equivalent.*

(4) *We have $T'(n) \succeq \exp(S'_M(n))$ for the generalized time function $T'(n)$ of the machine \mathcal{S} .*

Proof. (1) Assume that a word u belongs to the language \mathcal{L} recognizing by M . By Lemma 2.5, this word belongs to the language of $S(M)$, and the accepted computation C is positive. By Lemma 2.7(5), u belongs to the language of $\mathcal{L}_{\mathcal{S}}$ of \mathcal{S} .

Now suppose u belongs to $\mathcal{L}_{\mathcal{S}}$, and C is the accepted computation. Then the computation $C_{S(M)}$ is positive by Lemma 2.8, and therefore this computation is also an accepted computation of the machine M by Lemma 2.5, and so $u \in \mathcal{L}$.

(2) The above argument shows that if a reduced computation $C : w_0 \rightarrow \dots \rightarrow w_t$ of \mathcal{S} accepts an input admissible word w_0 , then the computation $C_{S(M)}$ is a positive accepted computation of both $S(M)$ and M , and so $S_M(n) \leq S_{\mathcal{S}}(n)$ by Lemma 2.7 (4). On the other hand, every accepted input configuration W of M is accepted by $S(M)$. By Lemma 2.7(5), it has a copy accepted by \mathcal{S} , and moreover, the accepting computations of M and \mathcal{S} need the same space. Therefore $S_M(n) \geq S_{\mathcal{S}}(n)$.

(3) Assume that $C : W = W_0 \rightarrow \dots \rightarrow W_t$ is an accepting computation of M such that $\text{space}_M(C) = \text{space}_M(W)$, and $C_{\mathcal{S}} : w = w_0 \rightarrow \dots \rightarrow w_s$. For given w and w_s , we also consider a reduced computation $C' : w \rightarrow \dots \rightarrow w_s$ of \mathcal{S} with minimal space and the computation $(C')_{S(M)} : W'_0 \rightarrow \dots \rightarrow W'_{t'}$. If $t' = 0$ (no basic rules), then $W_0 = W_t$ by Lemma 2.6 (2), and so $\text{space}_M(C) = \text{space}_{\mathcal{S}}(C')$. Then we assume that $t' > 0$ and

note that $\|W'_0\| = \|W_0\|$ and $\|W'_t\| = \|W_t\|$ by Lemma 2.6 (2) since the corresponding admissible words of \mathcal{S} can be connected by computations without basic rules. Since $C'_{S(M)}$ is positive by Lemma 2.8, it is also an accepted computation of the machine M by Lemma 2.5. Therefore, by Lemma 2.7 (4), $space_M(C) \leq space_{S(M)}(C'_{S(M)}) = space_{\mathcal{S}}(C')$. Since $|w|_a = |W|_a$, the last inequality proves that $S'_M(n) \leq S'_{\mathcal{S}}(n)$ for every n .

Now we consider any accepting computation $C : w = w_0 \rightarrow \dots \rightarrow w_t$ of \mathcal{S} with $|w| \leq n$, such that $space_{\mathcal{S}}(C) = space_{\mathcal{S}}(w)$. Without loss of generality, we may also assume that $space_{\mathcal{S}}(C[m]) = space_{\mathcal{S}}(w_m)$ for every subcomputation $C[m]$ of the form $w_m \rightarrow \dots \rightarrow w_t$. Let $\theta_{i_1}, \dots, \theta_{i_s}$ be the basic commands of the history $h = \theta_1 \dots \theta_t$ of C , C' be the subcomputation of C with history $h' = \bar{\theta}_1 \dots \bar{\theta}_{i_1-1}$, and C'' have history $h'' = \bar{\theta}_{i_1} \dots \bar{\theta}_t$; and so $h = h'h''$ and $C = C'C''$.

We denote by $W_0 \rightarrow W_1 \rightarrow \dots \rightarrow W_s$ the computation $(C'')_{S(M)}$ and by $(C'')_{S(M)}[1]$ the subcomputation $W_1 \rightarrow \dots \rightarrow W_s$, which is positive by Lemma 2.8. Note that $space_{S(M)}((C'')_{S(M)}[1]) = space_{S(M)}(W_1)$ since otherwise the subcomputation of $C[i_1]$ could be replaced by a subcomputation which needs less space by lemmas 2.7 (5). Therefore by Lemma 2.7 (4),

$$space_{\mathcal{S}}(C'') = space_{S(M)}((C'')_{S(M)}) \leq space_{S(M)}((C'')_{S(M)}[1]) + |||W_1||| - |||W_0||| = space_{S(M)}(W_1)$$

since $|||W_1||| - |||W_0|||$ is bounded by a constant c depending on the machine M only.

By Lemma 2.7 (2), we have $|w_j| \leq |w_0|$ for $j \leq i_1$, and so $space_{\mathcal{S}}(C') \leq |w_0|_a$. Now, since $|W_1| \leq |W_0| + c \leq |w_{i_1-1}| + c \leq |w_0| + c$, we get

$$\begin{aligned} space_{\mathcal{S}}(C) &\leq \max(space_{\mathcal{S}}(C'), space_{\mathcal{S}}(C'')) \leq \\ &\max(|w_0|_a, space_{S(M)}(W_1) + c) \leq \max(|w_0|_a, S'_{S(M)}(|w_0|_a + c) + c \end{aligned}$$

Hence $S'_{\mathcal{S}}(n) \leq \max(S'_{S(M)}(n + c) + c, n)$. This inequality together with the inequality $S'_M(n) \leq S'_{\mathcal{S}}(n)$ obtained earlier, show that $S'_M(n) \sim S'_{\mathcal{S}}(n)$.

(4) Let again $C : W = W_0 \rightarrow \dots \rightarrow W_t$ be an accepting computation of M such that $space_M(C) = space_M(W) = S'(|W|_a)$, and $C_S : w = w_0 \rightarrow \dots \rightarrow w_s$. For given w and w_s , we also consider a reduced computation $C' : w \rightarrow \dots \rightarrow w_s$ of \mathcal{S} with minimal time s and the computation $(C')_{S(M)} : W'_0 \rightarrow \dots \rightarrow W'_t$, where one has $W'_0 = W_0$ and $W'_t = W_t$. Therefore $(C')_{S(M)}$ is a positive computation by Lemma 2.8, and by the choice of C and Lemma 2.5, $space_M(C) \leq space_M((C')_{S(M)})$. Consider also a maximal subcomputation C'' of C' starting and ending with basic commands. Then $(C')_{S(M)} = (C'')_{S(M)}$. By Lemma 2.8, $space_{S(M)}(C'')_{S(M)} \leq c \log(time_{\mathcal{S}}(C''))$ for a constant $c > 0$. Thus by Lemma 2.5,

$$\begin{aligned} S'(|W|_a) &= space_M(C) \leq space_M((C')_{S(M)}) = space_{S(M)}((C')_{S(M)}) \\ &= space_{S(M)}((C'')_{S(M)}) \leq c \log(time_{\mathcal{S}}(C'')) \leq c \log(time_{\mathcal{S}}(C')) \leq c \log(T'(|W|_a)), \end{aligned}$$

and the lemma is proved. □

Lemma 2.10. (a) For every DTM M recognizing a language \mathcal{L} and having a space function $S(n)$, (b) for every NTM M recognizing a language \mathcal{L} and having an FSC space function $f(n)$, there exists an S -machine \mathcal{S} with the following properties.

1. The machine \mathcal{S} recognizes the language \mathcal{L} .

2. Respectively, (a) both the space and the generalized space functions of \mathcal{S} are equivalent to $S(n)$, (b) both the space and the generalized space functions of \mathcal{S} are equivalent to $f(n)^2$.
3. Every command of \mathcal{S} or its inverse inserts/deletes at most one letter on the left and at most one letter on the right of every state letter.
4. The machine \mathcal{S} satisfies the \vec{s}_{10} -property.
5. The unique start command is of the form $q_1 \rightarrow q'_1, q_2 \xrightarrow{\ell} q'_2, \dots, q_{k+1} \xrightarrow{\ell} q'_{k+1}$, where $(q_1, \dots, q_{k+1}) = \vec{s}_1$.
6. Any state letters q from Q_1 is passive, i.e., there are no commands of \mathcal{S} of the form $q \rightarrow q'u$ with non-empty a -word u .

Proof. For every DTM M recognizing a language \mathcal{L} and having a space function $S(n)$ (for every NTM M recognizing a language \mathcal{L} and having an FSC space function $f(n)$) one can construct a symmetric NTM M' , as in the formulation of Lemma 2.4. Then the properties 1 and 2 hold for the machine $\mathcal{S} = M' \circ Z$ by Lemma 2.9.

To provide the third property we use the same trick as for property 4 of Lemma 2.4. The properties (1) and (2) are obviously preserved.

To obtain the \vec{s}_{10} -condition of \mathcal{S} , it suffices to add new state letters and two new (positive) commands. Such a modification preserves the other properties of \mathcal{S} , as this was noticed at the end of subsection 2.1. Now the \vec{s}_{10} -condition for \mathcal{S} follows from Lemma 2.7 (1). The form of the start command follows from our agreement that all tapes number $2, \dots, k$ are empty for the input configurations of the machine M . Finally, the left-most head of the Turing machine M is passive being equal to the separating symbol α_1 ; and the same property is inherited by \mathcal{S} as this follows from the definition of composition $M' \circ Z$. \square

3 Groups and diagrams

3.1 Construction of embeddings

Let H be a finitely generated group with solvable word problem. To prove Theorem 1.2, we will suppose that a Turing machine M solves the word problem in H . This implies that H has a finite set of generators $\{a_1, \dots, a_m\}$, and a word w in generators a_i -s is accepted by M iff $w = 1$ in H . Here we consider only positive words in the generators since M can work with positive words only, and so we assume that the set of generator is symmetric: for every a_i , there is a generator a_j such that $a_i a_j = 1$ in H . Thus every relations of H follows from relations in a_1, \dots, a_m , with positive left-hand side.

Further we will assume that one of the hypotheses (a), (b) of Lemma 2.10 holds. Therefore we also have the S -machine $\mathcal{S} = \langle X, Y, Q, \Theta, \vec{s}_1, \vec{s}_0 \rangle$ provided by that Lemma. The input alphabet of \mathcal{S} is the system of generators a_1, \dots, a_m of the group H together with the symbols of the inverse letters $\{a_1^{-1}, \dots, a_m^{-1}\}$. Let $S'_S(n)$ be the generalized space function of \mathcal{S} .

We denote by $\hat{\mathcal{S}} = \langle \hat{X}, \hat{Y}, \hat{Q}, \hat{\Theta}, \hat{\vec{s}}_1, \hat{\vec{s}}_0 \rangle$ a copy of the S -machine \mathcal{S} . We will assume that $\hat{X} = X$, $Y \cap \hat{Y} = X$, $Q \cap \hat{Q}$ consists of the state letters of the vectors \vec{s}_1 , and \vec{s}_0 , and $\Theta \cap \hat{\Theta} = \emptyset$. Therefore the machines \mathcal{S} and $\hat{\mathcal{S}}$ have the same input admissible words.

The copy of a command θ of \mathcal{S} is called $\hat{\theta}$. Similar notation is used for the a -letters from \hat{Y} and q -letters from \hat{Q} . The set of rules of the machine and admissible words of $\mathcal{S} \cup \hat{\mathcal{S}}$ is by definition, the union of the corresponding sets for \mathcal{S} and for $\hat{\mathcal{S}}$. The following lemma is a clear consequence of these definitions.

Lemma 3.1. *The sets of the accepted input admissible words of the machines \mathcal{S} , $\hat{\mathcal{S}}$ and $\mathcal{S} \cup \hat{\mathcal{S}}$ coincide, and so these machines recognize the same language \mathcal{L} . They also have equal space functions and equal generalized space functions. For every accepting computation $w_1 \rightarrow \dots$ of $\mathcal{S} \cup \hat{\mathcal{S}}$, there is an accepting computation $w_1 \rightarrow \dots$ of either \mathcal{S} or $\hat{\mathcal{S}}$ whose length and space does not exceed the length and space of the original computation.*

We consider a group $G(\mathcal{S}, L)$ associated with the machine \mathcal{S} . Furthermore as in [21], we need a very similar group $\hat{G}(\mathcal{S}, L)$ to produce a group embedding required for the proof of Theorem 1.2.

To define $G(\mathcal{S}, L)$ we need many copies of every letter used in the work of \mathcal{S} ; this enables to apply a kind of hyperbolic argument for the hub structure of van Kampen diagrams. Moreover, the copies alternate with "mirror copies"; this trick works in Section 4.

Therefore we "multiply" the machine \mathcal{S} as follows. For some even $L \geq 40$, we introduce $L/2$ copies $\mathcal{S} = \mathcal{S}_1, \mathcal{S}_3, \dots, \mathcal{S}_{L-1}$ of the machine \mathcal{S} and $L/2$ mirror copies $\mathcal{S}_2, \mathcal{S}_4, \dots, \mathcal{S}_L$ of \mathcal{S} (A mirror copy of an arbitrary word $x_1 \dots x_n$ is, by definition, $x_n \dots x_1$. For even i , the rules of $\mathcal{S}_i(L)$ transform the words in the mirror manner in comparison with \mathcal{S} .) We also add auxiliary separating state letters k_1, \dots, k_L . For every admissible word W of the machine \mathcal{S} , we define the words $W = W_1, W_2, \dots, W_L$, where W_1, W_3, \dots, W_{L-1} are copies of W in disjoint alphabets and W_2, W_4, \dots, W_L are mirror copies of W also in disjoint alphabets. The words of the form $k_1 W_1 k_2 W_2 \dots k_L W_L$ are admissible word of the machine $\mathcal{S}(L)$. The rules of $\mathcal{S}(L)$ are in one-to-one correspondence with the rules of \mathcal{S} ; they transform the words W_1, W_3, \dots as the commands of \mathcal{S} , and transform the words W_2, W_4, \dots in the mirror manner, and they do not change the new state letters k_1, \dots, k_L . We identify the set of rules of $\mathcal{S}(L)$ with Θ . Thus, by definition of $\mathcal{S}(L)$, we also manifold the input and accept configurations. If u is an input word for \mathcal{S} , then we denote by $\Sigma(u, L)$ the corresponding input configuration for $\mathcal{S}(L)$ containing $L/2$ copies of u and $L/2$ mirror copies of u as subwords. If the admissible words of \mathcal{S} have K state letters, then the admissible words of the machine $\mathcal{S}(L)$ have $N = (K + 1)L$ state letters. Clearly the machine $\mathcal{S}(L)$ enjoys the properties (1)–(5) of machine \mathcal{S} listed in Lemma 2.10.

The finite set of generators of the group $G(\mathcal{S}, L)$ consists of q -letters corresponding to the states of $\mathcal{S}(L)$, a -letters corresponding to the tape letters of $\mathcal{S}(L)$, and θ -letters corresponding to the commands. Thus the set of generators consists of the set of (state) q -letters $Q(\mathcal{S}, L) = \sqcup_{i=1}^N Q_i$, the set of (tape) a -letters $Y = \sqcup_{i=1}^N Y_i$ including $X = \sqcup_{i=1}^N X_i$, and the θ -letters from N copies of Θ^+ , i.e., for every $\theta \in \Theta^+$, we have N generators $\theta_1, \dots, \theta_N$.

The relations of the group $G(\mathcal{S}, L)$ correspond to the rules of the machine $\mathcal{S}(L)$; for every $\theta = [U_1 \rightarrow V_1, \dots, U_N \rightarrow V_N] \in \Theta^+$, we have

$$U_i \theta_{i+1} = \theta_i V_i, \quad \theta_j a = a \theta_j, \quad i, j = 1, \dots, N \quad (3.4)$$

for all $a \in \bar{Y}_j(\theta)$. (Here $\theta_{N+1} = \theta_1$.) The first type of relations will be called (θ, q) -relations, the second type - (θ, a) -relations.

The definition of machine $\hat{\mathcal{S}}(L)$ is similar to that of $\mathcal{S}(L)$ but the admissible words are of the form $k_1\hat{W}_1k_2\hat{W}_2\dots k_L\hat{W}_L$, where every \hat{W}_i is obtained from W_i after replacement of every letter x by its copy \hat{x} , and for $i = 1$, we, in addition, delete all a -letters, i.e., the word \hat{W}_1 has no a -letters. (In other words, instead of the first copy of \mathcal{S} , we use the "machine" with the same state letters but having no tape letters.) In particular, the word $\hat{\Sigma}(u, L)$ is obtained from $\Sigma(u, L)$ by omitting of the first occurrence of the input word u . Again, it is obvious that properties (1) - (5) from Lemma 2.10 hold for the machine $\hat{\mathcal{S}}(L)$ as well. The relations of the group $\hat{G}(\mathcal{S}, L)$ are

$$\hat{U}_i\hat{\theta}_{i+1} = \hat{\theta}_i\hat{V}_i, \quad i = 1, \dots, N, \quad \hat{\theta}_j\hat{a} = \hat{a}\hat{\theta}_j \quad (3.5)$$

for all $\hat{a} \in Y_j(\hat{\theta})$ and $j \in [K + 1, N]$.

We will also use the combined S -machine $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$. Its admissible words are either the admissible words for $\mathcal{S}(L)$ or the admissible words for $\hat{\mathcal{S}}(L)$, the set of rules is the union of rules for $\mathcal{S}(L)$ and $\hat{\mathcal{S}}(L)$. Note that the machine $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$ does not satisfy the \vec{s}_{10} -condition.

The subwords of the form $(k_iW_ik_{i+1})^{\pm 1}$ (indexes modulo L) of the admissible words of the machine $\mathcal{S}(L)$ are said to be an i -sector words. Similarly one define i -sector words for the machines $\hat{\mathcal{S}}(L)$ and for $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$. (Recall that the q -letters of the 1-sector words of $\mathcal{S}(L)$, except for k_1 and k_2 , are identified with the corresponding letters of the original S -machine \mathcal{S} .) The state letters of the i -sector of $\mathcal{S}(L)$ are the letters from $\sqcup_{j=(i-1)(K+1)+1}^{i(K+1)} Q_j$, and the tape letters of the i -sector are the letters from $\sqcup_{j=(i-1)(K+1)+1}^{i(K+1)} Y_j$. Similarly we have state and tape letters of the i -sector for $\hat{\mathcal{S}}(L)$ and for $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$. By definition, the θ -letters θ_j and $\hat{\theta}_j$ with subscripts $j \in [(i-1)(K+1)+1, i(K+1)]$ are θ -letters of the i -sector of $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$. The state, tape and the θ -letters of the i -sector constitute the *alphabet* \mathcal{A}_i of the i -sector.

The group $G(\mathcal{S} \cup \hat{\mathcal{S}}, L)$ is given by the generators and relations of both groups $G(\mathcal{S}, L)$ and $\hat{G}(\mathcal{S}, L)$.

Finally, the required group G is given by the generators and relations of the group $G(\mathcal{S} \cup \hat{\mathcal{S}}, L)$ and one more additional relation, namely the *hub*-relation

$$\Sigma_0 = 1, \quad (3.6)$$

where $\Sigma_0 = \Sigma(L)$ is the accept word (of length N) of the machine $\mathcal{S}(L)$ (and of $\hat{\mathcal{S}}(L)$ as well).

Suppose an admissible word W' of $\mathcal{S}(L)$ is obtained from an admissible word W by an application of a rule $\theta : [U_1 \rightarrow V_1, \dots, U_N \rightarrow V_N]$. This implies that $W = U_1w_1U_2w_2\dots U_Nw_N$, where w_i is a word in the alphabet $Y_i(\theta)$, and therefore $W' = \theta_1^{-1}W\theta_1$ in $G(\mathcal{S}, L)$ by relations (3.4), since $\theta_1 = \theta_{N+1}$.

Now suppose u is a positive word in the alphabet $\{a_1, \dots, a_m\}$ vanishing in the group H . Then it is recognized by machine \mathcal{S} , and so the word $\Sigma(u)$ is accepted by $\mathcal{S}(L)$ and therefore it is conjugate of Σ_0 in the group $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$. Consequently, we have $\Sigma(u) = 1$ in G by (3.6). Similarly, $\hat{\Sigma}(u) = 1$ in G .

Recall that we identified the alphabet of 1-sector words of $\mathcal{S}(L)$ with the alphabet of \mathcal{S} . Therefore the word $\hat{\Sigma}(u)$ results from $\Sigma(u)$ after deleting the subword u in the alphabet

of generators of H . Hence the relations $\Sigma(u) = \hat{\Sigma}(u) = 1$ imply $u = 1$ in G . Since the language of accepted words for $\mathcal{S}(L)$ contains all the defining relations of H , we have obtained

Lemma 3.2. *The mapping $a_i \mapsto a_i$ ($i = 1, \dots, a_i$) induces a homomorphism of the group H to G .*

In Section 4 we show that this homomorphism is injective.

3.2 Minimal diagrams

As in [19], we enlarge the set of defining relations of the group G by adding some consequences of defining relations. Taking into account Lemma 3.2, we include all cyclically reduced relations of the group H generated by the set $\{a_1, \dots, a_m\}$, i.e., all non-empty cyclically reduced words in $\{a_1^{\pm 1}, \dots, a_m^{\pm 1}\}$ which are equal to 1 in the group H . These relations will be called *H-relations*.

We denote by G_1 the group given by all generators of the group G , by all H -relations, and by all defining relations of G except for the hub-relation (3.6).

Recall that a van Kampen *diagram* Δ over a presentation $P = \langle B \mid \mathcal{R} \rangle$ (or just over the group P) is a finite oriented connected and simply-connected planar 2-complex endowed with a labeling function $\phi : E(\Delta) \rightarrow B^{\pm 1}$, where $E(\Delta)$ denotes the set of oriented edges of Δ , such that $\phi(e^{-1}) \equiv \phi(e)^{-1}$. Given a cell Π of Δ , we denote by $\partial\Pi$ the boundary of Π ; similarly, $\partial\Delta$ denotes the boundary of Δ . The labels of $\partial\Pi$ and $\partial\Delta$ are defined up to cyclic permutations. An additional requirement is that the label of any cell Π of Δ is equal to (a cyclic permutation of) a word $R^{\pm 1}$, where $R \in \mathcal{R}$. Labels and lengths of paths are defined as for Cayley graphs.

The van Kampen Lemma states that a word W over the alphabet $B^{\pm 1}$ represents the identity in the group P if and only if there exists a diagram Δ over P such that $\phi(\partial\Delta) \equiv W$ ([15], Ch. 5, Theorem 1.1).

We will study diagrams over the groups G and G_1 . The edges labeled by state letters ($= q$ -letters) will be called *q-edges*, the edges labeled by tape letters ($= a$ -letters) will be called *a-edges*, and the edges labeled by the letters from Θ and $\hat{\Theta}$ ($= \theta$ -letters) are *θ -edges*. The cells corresponding to the relation (3.6) are called *hubs*, the cells corresponding to the relation (3.4) and (3.5) are called *(θ, q)-cells* if they involve q -letters, and they are called *(θ, a)-cells* otherwise. The cells corresponding to arbitrary relations of H are *H-cells*.

The obtained presentation and diagrams over it are graded by the ranks of defining words and cells as follows. The hubs are the cells of the highest rank, the rank of *(θ, q)-cells* is higher than the rank of *H-cells* (and in Lemma 5.1, *(θ, k_i)-cells*, with $i \neq 1, 2$ are higher than other *(θ, q)-cells*), and the *(θ, a)-cells* are of the lowest rank.

If Δ and Δ' are diagrams over G , then we say that Δ has a higher type than Δ' if Δ has more hubs, or the numbers of hubs are the same, but Δ has more cells which are next in the hierarchy, and so on.

Clearly the defined partial order on the set of diagrams satisfies the descending chain condition, and so there is a diagram having the smallest type among all diagrams with the same boundary label. Such a diagram is called *minimal*.

3.3 Bands and trapezia

From now on, we shall mainly consider minimal van Kampen diagrams. In particular the diagrams are reduced, i.e., they do not contain cells that have a common edge and are mirror images of each other. To study van Kampen diagrams over the groups G and G_1 we shall use bands and trapezia as in [25], [4].

Here we repeat some necessary definitions.

Definition 3.3. Let \mathcal{Z} be a subset of the set of generators \mathcal{X} of the group G . An \mathcal{Z} -band \mathcal{B} is a sequence of cells π_1, \dots, π_n in a van Kampen diagram such that

- Each two consecutive cells π_i and π_{i+1} in this sequence have a common edge e_i labeled by a letter from \mathcal{Z} .
- Each cell π_i , $i = 1, \dots, n$ has exactly two \mathcal{Z} -edges, e_{i-1} and e_i (i.e. edges labeled by a letter from \mathcal{Z}).
- If $n = 0$, then \mathcal{B} is just an \mathcal{Z} -edge.

The counterclockwise boundary of the subdiagram formed by the cells π_1, \dots, π_n of \mathcal{B} has the factorization $e^{-1}q_1fq_2^{-1}$ where $e = e_0$ is an \mathcal{Z} -edge of π_1 , $f = e_n$ is an \mathcal{Z} -edge of π_n . We call q_1 the *bottom* of \mathcal{B} and q_2 the *top* of \mathcal{B} , denoted $\mathbf{bot}(\mathcal{B})$ and $\mathbf{top}(\mathcal{B})$. Top/bottom paths and their inverses are also called the *sides* of the band. The \mathcal{Z} -edges e and f are called the *start* and *end* edges of the band. If $n \geq 1$ but $e = f$, then the \mathcal{Z} -band is called an \mathcal{Z} -annulus.

We say that an \mathcal{Z}_1 -band and an \mathcal{Z}_2 -band *cross* if they have a common cell and $\mathcal{Z}_1 \cap \mathcal{Z}_2 = \emptyset$.

We shall call an \mathcal{Z} -band *maximal* if it is not contained in any other \mathcal{Z} -band.

We will consider q -bands where \mathcal{Z} is one of the sets Q_i of state letters for the machine $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$, θ -bands for every $\theta \in \Theta$, and a -bands where $M = \{a\} \subseteq Y$.

The convention is that a -bands do not contain q -cells, and so they consist of (θ, a) -cells only.

The papers [18], [4], [21] contain the proof of the following lemma in more general setting. (In contrast to lemmas 6.1 [18] and 3.11 [21], we have no x -cells here.)

Lemma 3.4. *A minimal van Kampen diagram Δ over G_1 has no q -annuli, no θ -annuli, and no a -annuli. Every θ -band of Δ shares at most one cell with any q -band and with any a -band.*

If $W = x_1 \dots x_n$ is a word in an alphabet X , Y is another alphabet, and $\phi: X \rightarrow Y \cup \{1\}$ (where 1 is the empty word) is a map, then $\phi(W) = \phi(x_1) \dots \phi(x_n)$ is called the *projection* of W onto Y . We shall consider the projections of words in the generators of G onto $\Theta \sqcup \hat{\Theta}$ (all θ -letters map to the corresponding element of $\Theta \sqcup \hat{\Theta}$, all other letters map to 1), and the projection onto the alphabet $\{Q_1 \sqcup \dots \sqcup Q_N\}$ (every q -letter maps to the corresponding Q_i , other letters map to 1).

Definition 3.5. The projection of the label of a side of a q -band onto the alphabet $\Theta^{\pm 1}$ is called the *history* of the band. The projection of the label of a side of a θ -band onto the alphabet $\{Q_1, \dots, Q_N\}$ is called the *base* of the band. Similarly we can define the history of a word and the base of a word. The base of a word W is denoted by $\mathbf{base}(W)$. It will be convenient instead of letters Q_1, \dots, Q_N , in base words, to use representatives of these

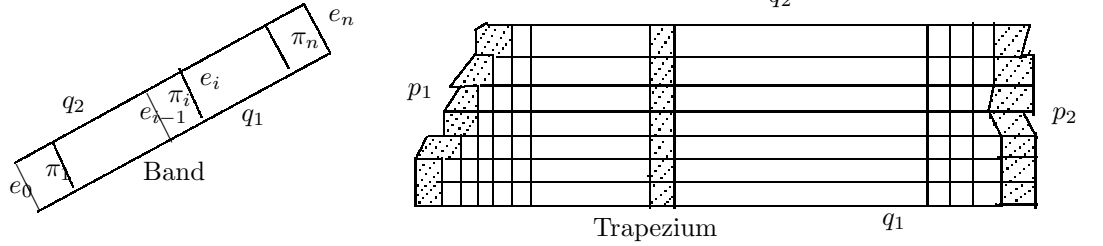
sets. For example, if $k \in Q_1$, $q \in Q_2$, we shall say that the word kaq has base kq instead of Q_1Q_2 .

Definition 3.6. Let Δ be a minimal van Kampen diagram over G_1 which has the contour of the form $p_1^{-1}q_1p_2q_2^{-1}$ where:

(TR_1) p_1 and p_2 are sides of q -bands,

(TR_2) q_1, q_2 are maximal parts of the sides of θ -bands such that $\phi(q_1), \phi(q_2)$ start and end with q -letters,

(TR_3) for every θ -band \mathcal{T} in Δ , the labels of **top**(\mathcal{T}) and **bot**(\mathcal{T}) are reduced.



Then Δ is called a *trapezium*. The path q_1 is called the *bottom*, the path q_2 is called the *top* of the trapezium, the paths p_1 and p_2 are called the *left and right sides* of the trapezium. The history of the q -band whose side is p_2 is called the *history* of the trapezium; the length of the history is called the *height* of the trapezium. The base of q_1 is called the *base* of the trapezium.

Remark 3.7. (1) Property (TR_3) is easy to achieve: by folding edges with the same labels having the same initial vertex, one can make the boundary label of a subdiagram in a van Kampen diagram reduced, see [25].

(2) Notice that the top (bottom) side of a θ -band \mathcal{T} does not necessarily coincide with the top (bottom) side q_2 (side q_1) of the corresponding trapezium of height 1, and q_2 (q_1) is obtained from **top**(\mathcal{T}) (resp. **bot**(\mathcal{T})) by trimming a first and last a -edges if these paths start or/and end with a -edges. We shall denote the trimmed top and bottom sides of \mathcal{T} by **ttop**(\mathcal{T}) and **tbot**(\mathcal{T}). By definition, for arbitrary θ -band \mathcal{T} , **ttop**(\mathcal{T}) is obtained by such a trimming only if \mathcal{T} starts or/and ends with a (θ, q) -cell; otherwise **ttop**(\mathcal{T}) = **top**(\mathcal{T}). The definition of **tbot**(\mathcal{T}) is similar.

The trapezium Δ is said to be a i -sector if the labels of top and bottom paths are i -sector words.

Lemma 3.8. Let Γ be an i -th sector, where $i \neq 1$. Then the sides p_1 and p_2 are the sides of maximal k_i - and k_{i+1} -bands \mathcal{K}_i and \mathcal{K}_{i+1} of Γ , respectively. If an edge e of Γ , belongs to neither \mathcal{K}_i nor \mathcal{K}_{i+1} , then $\phi(e) \in \mathcal{A}_i$. In particular, Γ has no H -cells.

Proof. The first assertion follows from Lemma 3.4, since the labels of the top and bottom of an i -sector are of the form $k_i \dots k_{i+1}$. Then, by the same lemma, the second assertion is true for the edges of all θ -cells since every maximal θ -band must connect \mathcal{K}_i and \mathcal{K}_{i+1} . Since $i \neq 1$ and the labels of the boundary edges of H -cells belong to \mathcal{A}_1 , the H -cells of Γ have no edges in common either with the θ -cells of Γ or with the boundary $\partial\Gamma$. Now

the minimality of the diagram Γ implies that Γ has no H -cells at all, and so the lemma is proved. \square

The following lemma claims that every i -sector, $i \neq 1$, simulates the work of $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$. It summarizes the assertions of lemmas 6.1, 6.3, 6.9, and 6.16 from [21]. For the formulation (1) below, it is important that \mathcal{S} (and $\mathcal{S} \cup \hat{\mathcal{S}}$) is an S -machine. The analog of this statement is false for Turing machines. (See [19] for discussion.)

Lemma 3.9. (1) *Let Δ be an i -sector for some $i \neq 1$ with history $\theta_1 \dots \theta_d$. Assume that Δ has consecutive maximal θ -bands $\mathcal{T}_1, \dots, \mathcal{T}_d$, and $k_i W_j k_{i+1}$, and $k_i W'_j k_{i+1}$ are the bottom and the top labels of \mathcal{T}_j , ($j = 1, \dots, d$). Let U_j (resp. V_j , $i = 1, \dots, d$) be the copies of W_j (resp. W'_j) in the alphabet of the machine $\mathcal{S} \cup \hat{\mathcal{S}}$. Then U_j, V_j are admissible words for $\mathcal{S} \cup \hat{\mathcal{S}}$, and*

$$V_1 \equiv U_1 \circ \theta_1, U_2 \equiv V_1, \dots, U_d \equiv V_{d-1}, V_d \equiv U_d \circ \theta_d$$

(2) *For every reduced computation $U \circ h \equiv V$ of \mathcal{S} (of $\hat{\mathcal{S}}$) with $|h| \geq 1$ and for every $i \in [1, N]$ (for every $i \in [2, N]$), there exists an i -sector Δ with history h and without H -cells, whose bottom and top labels are $k_i U' k_{i+1}$ and $k_i V' k_{i+1}$, where U' (resp. V') is the copy or the mirror copy of the word U (of V) in the alphabet \mathcal{A}_i . These copies are mirror ones iff i is even.*

\square

We call an i -sector *accepted* if the top label $k_i V' k_{i+1}$ of it is the i -sector subword of the word Σ_0 . For $i \neq 1$, the computation $V_1 \rightarrow \dots \rightarrow V_d$ of the machine $\mathcal{S} \cup \hat{\mathcal{S}}$ provided by Lemma 3.9 (1) for an accepted i sector is accepting.

3.4 Replicas

Let Δ be an i -sector, where $i \neq 1$. Then by Lemma 3.8, for every $i' \neq 1$, one can relabel the edges of Δ (or of the mirror copy of Δ if $i - i'$ is odd) and obtain an i' -sector Δ' , which is just a copy (or a mirror copy) of Δ . But one cannot construct such a copy if $i' = 1$ since there are no commutativity relations $\hat{\theta}_j \hat{a} = \hat{a} \hat{\theta}_j$ if $\hat{\theta}_j \in \mathcal{A}_1$ (see (3.5)). However one can construct an ersatz-copy of Δ called *replica* if Δ is an accepted sector. This construction will be used in Section 5.

For $i' = 1$, we construct the *replica* of Δ as follows. (We assume below that i is odd, otherwise one first replaces Δ by its mirror copy.)

At first, the relations (3.4) and (3.5) make possible to replace the maximal k_i -band \mathcal{K}_i and k_{i+1} -band \mathcal{K}_{i+1} of Δ by their copies \mathcal{K}_1 and \mathcal{K}_2 , respectively. Similarly, we replace every maximal θ -band of Δ by its copy if θ is a command of the machine \mathcal{S} . Now let \mathcal{T} be a maximal θ -band of Δ and θ a command of the machine $\hat{\mathcal{S}}$. This \mathcal{T} consists of (θ, q) - and (θ, a) -cells. To construct the replica \mathcal{T}' of \mathcal{T} we take the 'copies' of (θ, q) -cells only (but no a -edges in these 'copies', the a -edges are contracted to vertexes !) and build \mathcal{T}' from them identifying θ -edges of neighbor cells in the order in which the original (θ, q) -cells appear in \mathcal{T} .

It remains to close up the holes between θ -bands \mathcal{T}'_s and \mathcal{T}'_{s+1} for consecutive \mathcal{T}_s and \mathcal{T}_{s+1} . If the corresponding letters θ_s and θ_{s+1} are both the commands of \mathcal{S} , then we just identify the top of the copy \mathcal{T}'_s and the bottom of the copy \mathcal{T}'_{s+1} . This is possible since

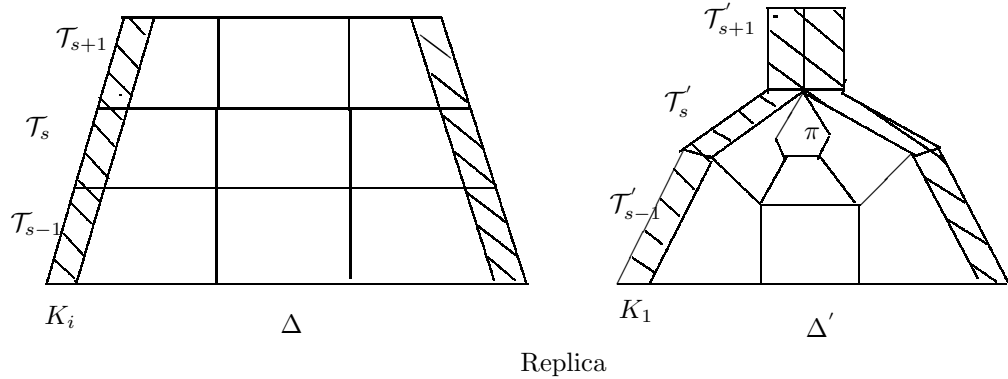
$\phi(\mathbf{top}\mathcal{T}_s) = \phi(\mathbf{bot}\mathcal{T}_{s+1})$ by Lemma 3.9 (1). Similar identification works if θ_s and θ_{s+1} are both the commands of $\hat{\mathcal{S}}$.

Assume now that θ_s is a command of \mathcal{S} and θ_{s+1} is a command of $\hat{\mathcal{S}}$ (or vice versa). Then $\phi(\mathbf{top}(\mathcal{T}_s)) = \phi(\mathbf{bot}(\mathcal{T}_{s+1})) = k_i W_s k_{i+1}$, and the copy W'_s of W_s in the alphabet \mathcal{A}_1 is an admissible word by both machines \mathcal{S} and $\hat{\mathcal{S}}$ since both commands θ_s^{-1} and θ_{s+1} are applicable to it. But the only common tape letters of these machines are the letters of the input alphabet of \mathcal{S} , and the only common state letters of these machines are the letters from the start vector \vec{s}_1 and the accept vector \vec{s}_0 of \mathcal{S} . Then by property (5) of Lemma 2.10, θ_s^{-1} is either the start rule or the accept rule of \mathcal{S} .

In the latter case, we have $\phi(\mathbf{top}(\mathcal{T}'_s)) = \phi(\mathbf{bot}\mathcal{T}'_{s+1})) = k_i W_s k_{i+1}$ since the accept words have no a -letters at all, and so the identification of the top of the copy \mathcal{T}'_s and the bottom of the copy \mathcal{T}'_{s+1} is possible again. Let us consider the former case. Then the rule θ_s^{-1} is of the form $q_1 \rightarrow q'_1, q_2 \xrightarrow{\ell} q'_2, \dots, q_K \xrightarrow{\ell} q'_K$, where $(q_1, \dots, q_K) = \vec{s}_1$. Therefore W'_s can have tape letters only between q_1 and q_2 , i.e., $W'_s = q_1 u q_2 q_3 \dots q_K$, where u is a word in the input alphabet.

Since the sector Δ is accepted and the rules of \mathcal{S} -machines are invertible, we have that W'_s is accepted by the machine $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$, and so it is accepted by \mathcal{S} by Lemma 3.1. Hence the word u belongs to the language recognized by the machine \mathcal{S} , and therefore u is a word in the generators of H , and $u = 1$ in H by the definition of that language.

Now, to close up the hole between the bands \mathcal{T}'_s and \mathcal{T}'_{s+1} , it suffices to paste in an H -cell π labeled by the cyclically reduced form of the word u between them because $\phi(\mathbf{top}(\mathcal{T}'_s)) = k_1 q_1 u q_2 q_3 \dots q_K k_2$ and $\phi(\mathbf{bot}\mathcal{T}'_{s+1}) = k_1 q_1 q_2 q_3 \dots q_K k_2$.



The replica Δ' of the accepted i -sector Δ is constructed. To summarize our effort: We cut off the (θ, a) -cell and contract up the a -edges of (θ, q) -cells for every command θ of the machine $\hat{\mathcal{S}}$ from Δ , replace the maximal k_i - and k_{i+1} -bands by their k_1 - and k_2 -copies, replace all the labels of the remaining edges by their copies from the alphabet \mathcal{A}_1 , and then close up all the holes by pasting in several H -cells. The result is the replica Δ' , canonically obtained above.

Remark 3.10. (1) The i -sector Δ is a union of $K + 1$ subsectors $\Gamma_1, \dots, \Gamma_{K+1}$, where every Γ_j is a trapezia with a base of length 2 and with height equal to the height of Δ . The subsector Γ_j has a common maximal q -band $\mathcal{C} = \mathcal{C}_{j+1}$ with Γ_{j+1} for $j = 1, \dots, K$. If i is odd (even), then Γ_2 (resp., Γ_K) is the *input subsector*. When we construct a replica, H -cells appear only in the input subsector of the replica.

Every Γ_j has its own replica Γ'_j , which is a subsector of Δ' . Similarly, every q - or θ -edge of \mathcal{C} has a replica in the replica \mathcal{C}' of \mathcal{C} in Δ' . Also every vertex o of \mathcal{C} belongs to either a q -edge or a θ -edge since the boundary of a (θ, q) -cell has no two consecutive a -edges by Lemma 2.10(5); and so o has a replica o' .

(2) The replica of an i -sector is not necessarily a minimal diagram.

3.5 Discs

A *disc diagram* (or a *disc*) is a (sub)diagram Δ such that (1) it has exactly one hub Π (2) there are no θ -edges on the boundary $\partial\Delta$ (3) There are no H -cells of Δ having an edge on $\partial\Delta$. In particular, a hub is a disc diagram.

Let us consider a disc diagram Δ with a hub Π . Denote by $\mathcal{K}_1, \dots, \mathcal{K}_L$ the k_1, \dots, k_L -bands starting on the hub Π . Since the hub relation has only one letter k_i for every i , these k -bands have to end on $\partial\Delta$. It therefore follows from Lemma 3.4 that for every i , the bands \mathcal{K}_i and \mathcal{K}_{i+1} bound, together with the $\partial\Delta$ and $\partial\Pi$, either a subdiagram Ψ_i having no cells corresponding to any non-trivial relation of G (in this case the bands \mathcal{K}_i and \mathcal{K}_{i+1} are also trivial), or Ψ_i is a trapezium, and the boundary label of Δ has exactly one letter from \mathcal{Q}_i for every $i \in [1, \dots, N]$.

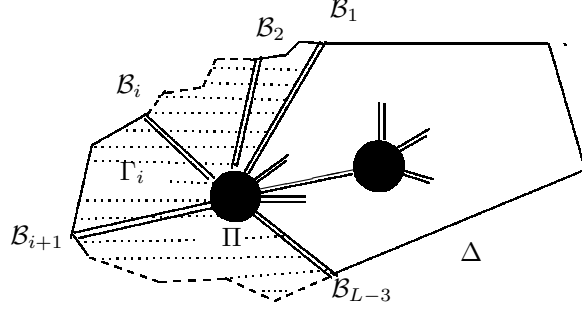
Similarly, consider two hubs Π_1 and Π_2 in a minimal diagram, connected by k_i -band \mathcal{K}_i and k_{i+1} -band \mathcal{K}_{i+1} , where $(i, i+1) \neq (1, 2)$, and there are no other hubs between these k -bands. These bands together with the $\partial\Pi_1$ and $\partial\Pi_2$, bound either a subdiagram Ψ_i having no cells, or Ψ_i is a trapezium. The former case is impossible since in this case the hubs have a common k_i -edge and they are mirror copies of each other contrary to the reducibility of minimal diagrams. We want to show that the latter case is not possible too.

Indeed, in the latter case Ψ_i is an accepted trapezium since the i -sector subword of Σ_0 is $k_i w k_{i+1}$, where w is a (mirror) copy of the accept word of the machine \mathcal{S} . Therefore, according to Subsection 3.4, a replica Ψ_1 of Ψ_i (as well as the (mirror) copies Ψ_j for every $j \in [2, \dots, N]$) can be constructed. Then one can construct a spherical diagram Γ from Π_1 , Π_2 , and the diagrams Ψ_j ($j = 1, \dots, N$). There are two subdiagrams Γ' and Γ'' of Γ with common boundary: Γ' , being a copy of a subdiagram of the original diagram, is made of Π_1 , Π_2 , and $\Psi = \Psi_i$, and Γ'' is a union of all Ψ_j -s with $j \neq i$. Hence the subdiagram Γ' of the original diagram could be replaced by diagram of lower type with the same boundary label because Γ'' has no hubs. This contradicts the minimality of the original diagram.

Thus, any two hubs of a minimal diagram are connected by at most two k -bands, such that the subdiagram bounded by them contain no other hubs. This property makes the hub graph of a minimal diagram (where maximal k -bands play the role of edges connecting hubs) hyperbolic (in a sense) since the degree L of every vertex (=hub) is high (≥ 40). Below we give more precise formulation (proved for diagrams with such hub graph, in particular, in [18], Lemma 3.2).

Lemma 3.11. *If a minimal diagram over the group G contains a least one hub, then there is a hub Π in Δ such that $L - 3$ consecutive maximal k -bands $\mathcal{B}_1, \dots, \mathcal{B}_{L-3}$ start on Π , end on the boundary $\partial\Delta$, and for any $i \in [1, L - 4]$, there are no discs in the subdiagram Γ_i bounded by \mathcal{B}_i , \mathcal{B}_{i+1} , $\partial\Pi$, and $\partial\Delta$. \square*

Corollary 3.1. *The canonical homomorphism $H \rightarrow G$ given by Lemma 3.2 is injective.*



Proof. Assume that a word w in the generators a_1, \dots, a_m of the group H is equal to 1 in G . Then by van Kampen's Lemma, there is a minimal diagram Δ over G whose boundary label is w . Since w has no q -letters, Δ has no hubs by Lemma 3.11. By Lemma 3.4, Δ contains neither q - nor θ -annuli, and so it has neither (θ, q) -cells nor (θ, a) -cells, because w has neither q - nor θ -letters. Hence this diagram can contain H -cells only. Since the boundary labels of H -cells are trivial in H , the boundary label w is trivial in H too by van Kampen's lemma, and so the homomorphism is injective. \square

4 Comparison of paths in diagrams

4.1 Paths in sectors

We will modify the length function on the words in the generators of the group G . This modification is helpful in subsequent subsections.

The standard length $||*||$ of a word (of a path) will be called its *combinatorial length*. From now on we use the word *length* for the modified length. We set the length of every q -letter equal 1, and the length of every a -letter equal a small enough number $\delta > 0$ so that

$$\delta < (3N)^{-1}. \quad (4.7)$$

If a word v has s θ -letters, t a -letters, and no q -letters, then

$$|v| = s + \delta \max(0, t - s)$$

by definition. For example, the word read between two q -letters of a (q, θ) -relation has length 1 since it has one θ -letter and at most one a -letter by formulas (3.4, 3.5) and the property (3) of Lemma 2.10.

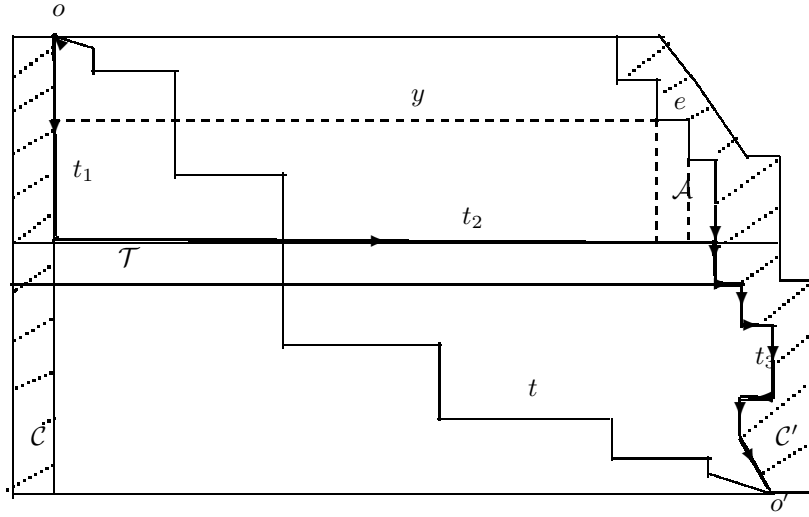
Arbitrary word w is a product $v_0 q_1 v_1 q_2 \dots q_m v_m$, where q_1, \dots, q_m are q -letters and the words v_0, \dots, v_m have no q -letters. Then, by definition, $|w| = m + \sum_{j=0}^m |v_j|$. The *length of a path* in a diagram is the length of its label. The *perimeter* $|\partial\Delta|$ of a van Kampen diagram is similarly defined by a shortest cyclic decompositions of the boundary $\partial\Delta$. It follows from this definition that for any product $s = s_1 s_2$ of two words or paths, we have $|s| \leq |s_1| + |s_2|$, and $|s| = |s_1| + |s_2|$ if s_2 starts or s_1 ends with a q -letter.

If a path p starts at a vertex o and ends at o' , we will write $o = p_-$ and $o' = p_+$.

Lemma 4.1. *Let Δ be a trapezium bounded by two maximal q -bands \mathcal{C} and \mathcal{C}' and having no H -cells. Assume that \mathcal{C} has no a -edges. Let o be vertex lying on both \mathcal{C} and the top of the trapezium Δ , and o' belong \mathcal{C}' and the bottom of Δ . Assume that a path t connects o and o' and has no q -edges. Then the vertexes o and o' can be connected in Δ by a path t' such that $|t'| \leq |t|$ and $t' = t_1 t_2 t_3$ where t_1 and t_3 are parts of the sides of \mathcal{C} and \mathcal{C}' , respectively, and t_2 consists only of a -edges.*

Proof. Since the path t has no q -edges, it follows from the assumption of the lemma that every maximal θ -band \mathcal{T} of Δ has exactly two (θ, q) -cells (the first one and the last one). Therefore \mathcal{C} and \mathcal{C}' can be connected along \mathcal{T} by a path x consisting of a -edges only.

We denote by t_2 a shortest path among such x -s. Then we define t_1 (t_3) as the shortest subpath of the side of \mathcal{C} (of \mathcal{C}') connecting o and $(t_2)_-$ ($(t_2)_+$ and o').



Assume that there is an a -bands \mathcal{A} starting with an a -edge of t_2 and ending with an a -edge e of $\partial\mathcal{C}'$. Then e belongs to some path ye where y consists of a -edges and connects \mathcal{C} and \mathcal{C}' . Notice that every maximal a -band crossing the path y must cross t_2 because it cannot cross \mathcal{A} , and \mathcal{C} has no a -edges. Hence $|y|_a \leq |t_2|_a - 1$, contrary the minimality in the choice of t_2 .

Thus every maximal a -band \mathcal{A} crossing t_2 must connect the top and the bottom of the trapezium Δ , and therefore the path t must cross every such an a -band \mathcal{A} . Also t must cross every maximal a -band starting on t_3 whence $|t|_a \geq |t_2|_a + |t_3|_a = |t'|_a$. Since the path $|t|$ must cross every maximal θ -band of Δ we also have inequality $|t|_\theta \geq |t_1|_\theta + |t_3|_\theta = |t'|_\theta$. Now it follows from the definition of path length that $|t'| \leq |t|$ as required. \square

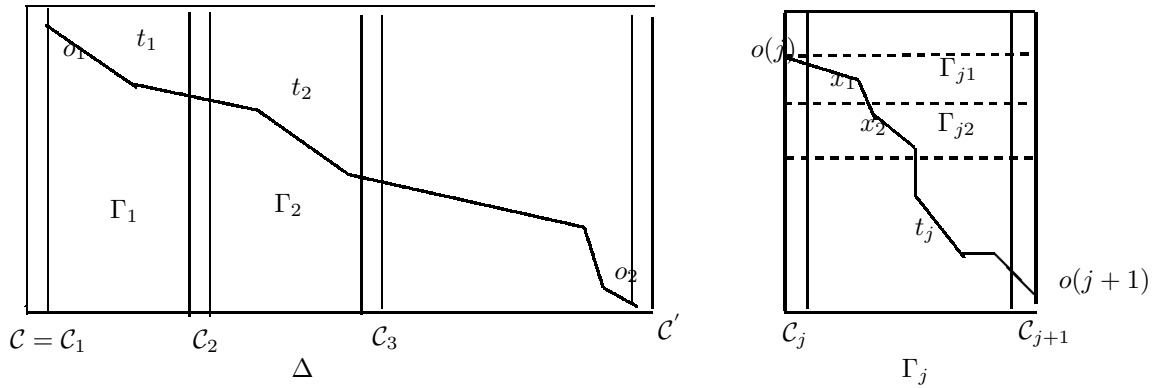
Lemma 4.2. *Let Δ be an accepted i -sector with index $i \neq 1$ bounded by two maximal k_i -band \mathcal{C} and k_{i+1} -band \mathcal{C}' . Let o_1 and o_2 be two vertexes lying on \mathcal{C} and \mathcal{C}' , respectively. Assume that o_1 and o_2 are connected by a path t , in Δ . Then the replicas o'_1 and o'_2 of the vertexes o_1 and o_2 in the replica Δ' of Δ can be connected by a path t' such that $|t'| \leq |t|$.*

Proof. First of all, one may assume that no one maximal q -band $\mathcal{C} = \mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{k+2} = \mathcal{C}'$ is crossed by the path t twice. Indeed, otherwise t has a subpath s of the form ezf , where e and f are q -edges of some \mathcal{C}_j separated in this band by m (θ, q) -cells for some $m \geq 0$. Therefore the path z must cross at least m maximal θ -bands whence $|ezf| \geq m + 2$. But the vertexes e_- and f_+ can be connected along \mathcal{C}_j by a path of length m (see the example after the definition of length $|\ast|$), and so the path t can be shortened.

Thus the path t is a product $t = t_1 \dots t_{k+1}$, where each t_j connects a vertex $o(j)$ lying on \mathcal{C}_j with a vertex $o(j+1)$ lying on \mathcal{C}_{j+1} , and for every $j = 1, \dots, k$, either t_{j+1} starts or t_j ends with a q -edge, and so $|t| = \sum_{j=1}^{k+1} |t_j|$. As in the previous paragraph, we have that each of t_j -s crosses every θ -band at most once. (Consider ezf , where e and f are θ -edges of the same θ -band.) Now using notation of Remark 3.10, it suffices to consider the replica Γ'_j of the subsector Γ_j and find a path t'_j connecting the replicas $o'(j)$ and $o'(j+1)$, with $|t'_j| \leq |t_j|$.

We may assume that i is odd. (If i is even one should use a mirror argument.)

We first consider the path t_2 crossing the input subsector Γ_2 , assuming that t_2 has no q -edges, since the q -edges (if any) can be attributed to the subpaths t_1 and t_3 . By property (6) of Lemma 2.10, \mathcal{C}_2 has no a -edges. Hence, by Lemma 4.1 applied to a subtrapezium of Γ_2 containing t_2 , we may assume that $t_2 = s_1 s_2 s_3$, where s_1 and s_3 are the subpaths of top or bottom paths of q -bands \mathcal{C}_2 and \mathcal{C}_3 , respectively, and s_2 goes along a top or bottom of a maximal θ -band \mathcal{T} . For the both paths s_1 and s_3 we have paths s'_1 and s'_3 of the same length lying on the boundaries of the q -bands \mathcal{C}'_2 and \mathcal{C}'_3 of the replica Γ'_2 and connecting the replicas of the vertexes $(s_1)_\pm$ and $(s_3)_\pm$, respectively. The vertexes $(s'_1)_+$ and $(s'_3)_-$ are either connected by a copy s'_2 of s_2 (if the θ -band \mathcal{T} was copied when we constructed the replica Δ') or $(s'_1)_+ = (s'_3)_-$ (if the corresponding θ -band of Δ' has no a -edges). It follows that in any case we have $|t'_2| \leq |t_2|$ for $t'_2 = s'_1 s'_2 s'_3$.



Assume now that $j \neq 2$. The subsector Γ_j has no H -cells by Lemma 3.8, and so it is a union of alternating subtrapezia $\Gamma_{j1}, \Gamma_{j2}, \dots$ whose histories are words either in the alphabet Θ or in $\hat{\Theta}$. Let $t_j = x_1 \dots x_d$, where every x_k belongs to some Γ_{js} . If the history of the trapezia Γ_{js} is a word over Θ , then we have the copy Γ'_{js} of Γ_{js} in Δ' , and a subpath $x = x_k$ of t_j lying in Γ_{js} has a copy x' in Γ'_{js} . If the history is a word over $\hat{\Theta}$, then Γ'_{js} has no a -edges, and for every subpath x of t_j lying in Γ_{js} , we can construct a corresponding subpath x' in Γ'_{js} which copies only q - and θ -edges of x , but ignores the

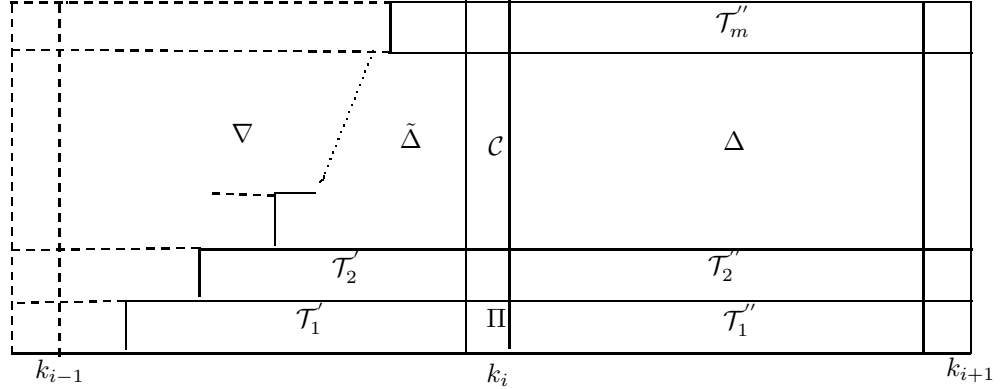
a -edges of x . Since there are no a -edges in the common boundaries of neighbor Γ'_{js} and $\Gamma'_{j,s+1}$, we have $(x'_k)_+ = (x'_{k+1})_-$ for every $k = 1, \dots, d-1$, and we obtain $|t'_2| \leq |t_2|$ for the path $t'_2 = x'_1 \dots x'_d$.

Now the required path t' is obtained, and the lemma is proved. \square

Assume that we have a minimal diagram Γ with a cyclically reduced boundary label over the group $G(\mathcal{S} \cup \hat{\mathcal{S}}, L)$, which is separated by a maximal k_i -band (or k_{i+1} -band) \mathcal{C} in two parts $\tilde{\Delta}$ and Δ such that $\tilde{\Delta}$ is an accepted i -sector. Assume that every maximal θ -band $\mathcal{T}_1, \dots, \mathcal{T}_m$ of Γ crosses $\tilde{\Delta}$, the bottom x of \mathcal{T}_1 is a part of the boundary $\partial\Gamma$, and $\phi(x)$ is a suffix of the subword $k_{i-1} \dots k_i \dots k_{i+1}$ (a prefix of the subword $k_i \dots k_{i+1} \dots k_{i+2}$, respectively) of the word Σ_0 . Also we assume that for every $j \in [2, m]$, the trimmed bottom $\mathbf{tbot}\mathcal{T}_j$ is a subpath of the top $\mathbf{top}\mathcal{T}_{j-1}$. Below we call such a diagram *unfinished* if $i, i-1 \neq 1$ (if $i, i+1 \neq 1$, respectively).

Lemma 4.3. *The part $\tilde{\Delta}$ of the unfinished diagram Γ can be embedded in an $i-1$ -sector (respectively, $i+1$ -sector) ∇ which is a mirror copy of the i -sector Δ .*

Proof. Without loss of generality we assume that \mathcal{C} is a k_i -band in the definition of unfinished diagram Γ . We construct ∇ by induction on the length m of the history $h = h_1 \dots h_m$ of the i -sector Δ .



Let Π be the intersection cell for \mathcal{T}_1 and \mathcal{C} . Then \mathcal{T}_1 consists of Π and two subbands \mathcal{T}'_1 and \mathcal{T}''_1 , where \mathcal{T}'_1 belongs to $\tilde{\Delta}$ and \mathcal{T}''_1 belongs to Δ . Let $\Pi(1)'$ and $\Pi(1)''$ be the neighbor cells for Π in \mathcal{T}'_1 and in \mathcal{T}''_1 , respectively. Then $\Pi(1)'$ (if it exists) is a mirror copy of $\Pi(1)''$ with boundary label in the alphabet \mathcal{A}_{i-1} since these cells are determined by the same history h_1 and mirror q -letters of the word Σ_0 as this follows from relations (3.4, 3.5). Similar argument shows that if $\Pi'(1)$ has a neighbor $\Pi''(2)$ (where $\Pi'(2) \neq \Pi$) in \mathcal{T}'_1 , then the cell $\Pi''(1)$ has a neighbor $\Pi''(2)$ (where $\Pi''(2) \neq \Pi$) in \mathcal{T}''_1 , and $\Pi'(2)$ is a mirror copy of $\Pi''(2)$ with boundary label in alphabet \mathcal{A}_{i-1} . By induction, we obtain that \mathcal{T}'_1 is a mirror copy of a subband of \mathcal{T}''_1 starting with Π . Hence one can extend \mathcal{T}'_1 (and the subdiagram $\tilde{\Delta}$) and obtain a θ -band \mathcal{T}_1^∇ which is a mirror copy of \mathcal{T}''_1 .

Since the trimmed bottom of the band \mathcal{T}_2 is a subpath of the top of \mathcal{T}_1 , one can similarly subdivide \mathcal{T}_2 in $\Pi(2)$, \mathcal{T}'_2 , \mathcal{T}''_2 , and prove that \mathcal{T}'_2 is the mirror copy of a subband of \mathcal{T}''_2 , starting with the cell $\Pi(2)$ and having the boundary label over the alphabet \mathcal{A}_{i-1} . (\mathcal{T}'_2 and \mathcal{T}''_2 may include (θ, q) -cells and also (θ, a) -cells.) Therefore there is an extension \mathcal{T}_2^∇ .

of \mathcal{T}'_2 , and this extension is a mirror copy of \mathcal{T}''_2 . Then by induction we construct θ -bands $\mathcal{T}_3^\nabla, \dots, \mathcal{T}_m^\nabla$, and these θ -bands together with the q -band \mathcal{C} form the required $i-1$ -sector ∇ . \square

Lemma 4.4. *Let Δ be a minimal diagram with boundary path $x_1y_1x_2y_2$, where*

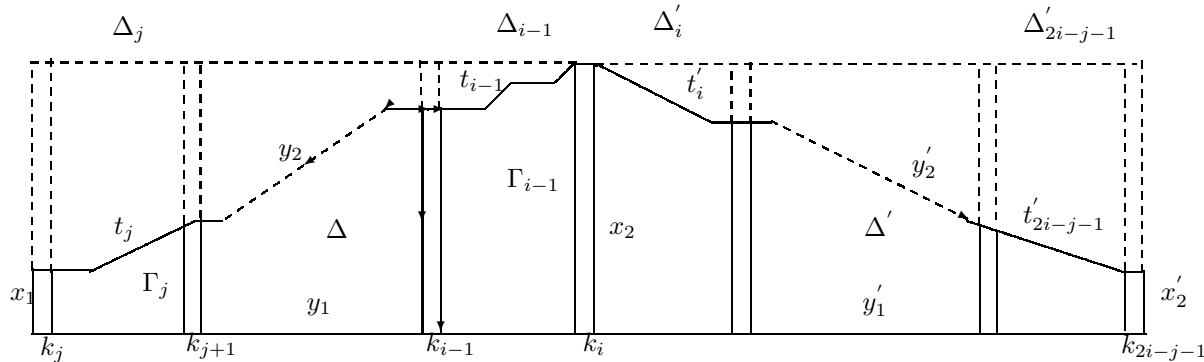
- (1) $\phi(y_1)$ is a subword $k_j \dots k_i$ of Σ_0 , this subword does not contain the letter k_1 ;
- (2) x_1 and x_2 are sides of the maximal k_j -band \mathcal{K}_j and k_i -band \mathcal{K}_i starting on y_1 ;
- (3) every cell of Δ belongs to one of the maximal θ -band $\mathcal{T}_1, \dots, \mathcal{T}_m$ of Δ ;
- (4) (a) either each of the bands $\mathcal{T}_1, \dots, \mathcal{T}_m$ crosses \mathcal{K}_j or (b) each of them crosses \mathcal{K}_i ;
- (5) the trimmed bottom path of \mathcal{T}_1 is a subpath of $y_1^{\pm 1}$, and the trimmed bottom path of \mathcal{T}_l is a subpath of the top path of \mathcal{T}_{l-1} for every $l = 2, \dots, m$;
- (6) one can construct a diagram $\bar{\Delta}$ with boundary $\bar{x}_1\bar{y}_1\bar{x}_2\bar{y}_2$, and $\bar{\Delta}$ satisfies the analogs of properties (1)-(5), but $\phi(\bar{y}_1) = k_{j-1} \dots k_i$ in case 4(a) ($\phi(\bar{y}_1) = k_j \dots k_{i+1}$ in case 4(b)), and Δ is embeddable in $\bar{\Delta}$ so that the k_j -band \mathcal{K}_j and the k_i -band \mathcal{K}_i remain maximal in $\bar{\Delta}$.

Then in case (4)(a) (in case 4(b)), there exists a diagram Δ' over the group G_1 with boundary path $x'_1y'_1x'_2y'_2$ such that $\phi((y'_1)^{-1})$ is the subword $k_{2j-i} \dots k_j$ (respectively, $k_i \dots k_{2i-j}$) of Σ_0 , x'_1 and x'_2 are sides of the maximal k_j -band \mathcal{K}_j and k_{2j-i} -band \mathcal{K}_{2j-i} (of the maximal k_i -band \mathcal{K}_i and k_{2i-j} -band \mathcal{K}_{2i-j} starting on y'_1), the label of x'_1 and x'_2 are copies of $\phi(x_1)$ and $\phi(x_2)$, resp., and $|y'_2| \leq |y_2|$.

(The subscripts of k -bands are taken modulo L .)

Proof. We consider the case (4)(b) only. The maximal k -bands $\mathcal{K}_j, \mathcal{K}_{j+1} \dots \mathcal{K}_{i+1}$ subdivide diagram $\bar{\Delta}$ in subdiagrams Γ_l -s, where Γ_l is bounded by \mathcal{K}_l and \mathcal{K}_{l+1} and Γ_l includes these k -bands ($l = j, \dots, i$).

Every maximal θ -band of Δ having a cell in Γ_{i-1} , must cross both bands \mathcal{K}_i and \mathcal{K}_{i+1} of $\bar{\Delta}$. Therefore the parts of these bands in Γ_{i-1} and Γ_i form an unfinished diagram whose i -sector is a subdiagram Δ_i of Γ_i . By Lemma 4.3, the subdiagram Γ_{i-1} is embedded into the mirror copy Δ_{i-1} of Δ_i . Similarly, Γ_{i-2} is embeddable into a mirror copy of Δ_{i-1} which is the copy of Δ_i (we denote this copy by Δ_{i-2}), ..., Γ_j is embeddable into the copy (or mirror copy) Δ_j of Δ_i .



Let $y_2 = t_{i-1} \dots t_j$, where t_l passes through Γ_l (and Δ_l) for $l = i-1, \dots, j$. Since every subpath t_l connects two vertexes on the k -bands of the l -sector Δ_l , we can construct

the mirror copy Δ'_{2i-l-1} (which is a $2i-l+1$ -sector) of Δ_l or the replica of Δ_l (if $2i-l-1=1$), and the copies of the vertexes $(t_l)_\pm$ are connected in Δ'_{2i-l-1} by a path t'_{2i-l-1} with $|t'_{2i-l-1}| \leq |t_l|$ by Lemma 4.2. The desired diagram Δ' embeds in the union of these Δ'_{2i-l-1} -s, and $y'_2 = t'_i \dots t'_{2i-j-1}$. \square

4.2 Shortcuts

In this subsection we show that Lemma 4.4 helps cutting off a hub from a diagram using a 'shortcut'. But first consider few simpler statements.

Lemma 4.5. *Let a diagram Δ over G have a q -band \mathcal{C} starting and ending on $\partial\Delta$, and p is a side of \mathcal{C} . Assume that no θ -band crosses \mathcal{C} twice in Δ and there is a factorization xy of the boundary path of Δ such that $x_- = p_-$, $x_+ = p_+$. Then $|p| \leq |x|$, $|\partial\mathcal{C}| \leq |\partial\Delta|$, and $|xp^{-1}| \leq |\partial\Delta|$.*

Proof. On the one hand, the length $|p|$ of p is equal to the number m of θ -cells in \mathcal{C} , since the every cell of \mathcal{C} has one θ -edge and at most one a -edge on p by condition 3 of Lemma 2.10 and the definition of (θ, q) -relations. On the other hand, every maximal θ -band crossing \mathcal{C} must terminate on x , since it does not cross \mathcal{C} twice. It follows that $|x| \geq m$, and so $|p| \leq |x|$. Similarly, we obtain inequalities $|\partial\mathcal{C}| \leq |\partial\Delta|$, and $|xp^{-1}| \leq |\partial\Delta|$. (We take into account the definition of length and the fact that the sides of the band \mathcal{C} are separated by two q -edges lying on $\partial\Delta$.) \square

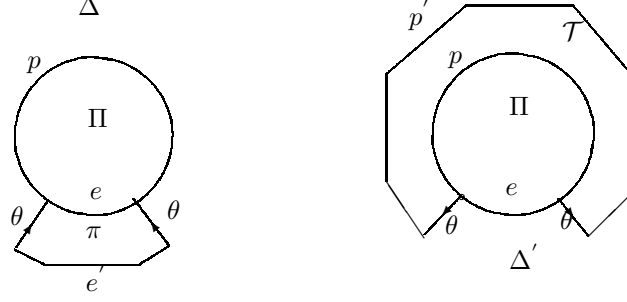
A maximal θ -band of a diagram Δ called a *rim band* if its start and end θ -edges as well as its top or its bottom lies on the boundary path of Δ .

Lemma 4.6. *Let Δ be a diagram over G with a rim band \mathcal{T} having at most N (θ, q) -cells. Denote by Δ' the subdiagram $\Delta \setminus \mathcal{T}$. Then $|\partial\Delta| - |\partial\Delta'| \geq 1/2$.*

Proof. Let s be the top side of \mathcal{T} and $s \subset \partial\Delta$. Note that by our assumptions the difference between the number of a -edges in the bottom s' of \mathcal{T} and the number of a -edges in s cannot be greater than $2N$ since every (θ, q) -cell has at most two a -edges. However, Δ' is obtained by cutting off \mathcal{T} along s' , and its boundary contains two θ -edges fewer than Δ . Thus one can compare the boundaries of Δ and Δ' as follows. There is a one-to-one correspondence between the q -edges of these boundaries, and to extend this correspondence to the θ - and a -edges of intermediate subpaths of the boundaries, one should remove 2 θ -edges from $\partial\Delta$ and add at most $2N$ a -edges. Therefore it follows from the definition of length and inequality (4.7), that $|\partial\Delta| - |\partial\Delta'| \geq 2(1 - \delta) - 2N\delta > 1/2$. \square

Lemma 4.7. *Let a diagram Δ have two cells: an H -cell Π and a (θ, a) -cell π which have a common a -edge e , ep is the boundary of Π and $efe'f'$ is the boundary of π , where $\phi(e) = \phi(e')^{-1} = a$ and $\phi(f) = \phi(f')^{-1} = \theta$ for a θ -letter θ . Then there is a diagram Δ' with the same boundary label as Δ composed of Π and a θ -band \mathcal{T} , and p is the side of \mathcal{T} .*

Proof. The letter a commute with θ , and therefore every letter of the boundary label of Π commutes with θ . Hence one can construct a θ -band \mathcal{T} with boundary $g'p^{-1}gp'$, where $\phi(p') = \phi(p)$, and $\phi(g) = \phi(g')^{-1} = \theta^{-1}$. If we attach the band \mathcal{T} to Π along the path p and remove π , we obtain the required diagram Δ' . \square



Let Π be a hub of a minimal diagram Δ given by Lemma 3.11. Using the notation of that lemma, we recall that the subdiagrams Γ_i and Γ_{i+1} intersect along the k -band \mathcal{B}_{i+1} ($i = 1, \dots, L-5$). We denote by Ψ the minimal subdiagram containing all the Γ_i -s for $i = 1, \dots, L-4$. The boundary path of Ψ is $x'x''$, where x' is composed from the sides of \mathcal{B}_1 , \mathcal{B}_{L-3} , and a subpath of $\partial\Pi$, while x'' is a subpath of $\partial\Delta$.

Lemma 4.8. *One can construct a minimal diagram Ψ' over G_1 with boundary path $x'\bar{x}$ such that (1) Ψ' includes the bands \mathcal{B}_1 and \mathcal{B}_{L-3} (2) $|\bar{x}| \leq |x''|$, (3) the subdiagram Ψ' has no maximal q -bands except for the q -bands \mathcal{B}'_i ($i = 1, \dots, L-3$) starting on x' , (4) every maximal θ -band of Ψ' crosses either the band $\mathcal{B}'_1 = \mathcal{B}_1$ or the band $\mathcal{B}'_{L-3} = \mathcal{B}_{L-3}$, (5) the subdiagram Ψ' has no H -cells between the pair of k -bands \mathcal{B}'_i and \mathcal{B}'_{i+1} , unless this pair is a pair of k_1 - and k_2 -bands.*

Proof. Let Ψ' be a minimal diagram with boundary of the form $x'\bar{x}$ which satisfies conditions (1) and (2) and has minimal $|\bar{x}|$. Since Ψ satisfies conditions (1) and (2) with $\bar{x} = x'$, such Ψ' exists. Clearly the path \bar{x} has no loops. If the diagram Ψ' has a maximal q -band \mathcal{C} which does not start/terminate on Π , then one can cut off \mathcal{C} and shorten \bar{x} by Lemma 4.5. So Ψ' satisfies condition (3).

Assume that the diagram Ψ' does not satisfy condition (4) of the lemma. Then by Lemma 3.4, we have a θ -band of Ψ' starting and terminating on \bar{x} . It follows that there is a θ -band \mathcal{T} starting and terminating on \bar{x} , such that the subdiagram Φ bounded by \mathcal{T} and a part y of \bar{x} has no non-trivial θ -bands, i.e., it contains only H -cells. Two H -cells of Φ cannot have a common edge since otherwise they can be replaced by one H -cell contrary to the minimality of Ψ' . It follows that every H -cell π of Φ has a common edge with \mathcal{T} because the path \bar{x} has no loops.

Thus we have a series of H -cells π_1, \dots, π_s in Φ with boundaries $y_i z_i$ ($i = 1, \dots, s$), where y_i is a part of y (or y_i is empty) and z_i belongs to the side z of \mathcal{T} . If $\sum |\phi(z_i)|_a \leq 2$, then $\sum |z_i| \leq 2\delta$ since every z_i is a product of a -edges. Therefore $|z| \leq |\bar{x}| + 2\delta$. It follows from Lemma 4.6 that if we remove all π_i -s and then cut off the band \mathcal{T} , then we decrease the length of \bar{x} since $2\delta < 1/2$; a contradiction.

Hence $\sum |\phi(z_i)|_a \geq 3$, and so at least one of π_i -s has a common edge with a (θ, a) -cell of \mathcal{T} . (We recall that \mathcal{T} intersects each of the maximal q -bands of Ψ' starting on x'' at most once by Lemma 3.4, and so at most two of the (θ, q) -cells of \mathcal{T} have a -edges with $a \in \mathcal{A}_1$, and each of these two (θ, q) -cells can have at most one a -edge.) Therefore we can

apply Lemma 4.7 to replace the (θ, a) -cell by a θ -band passing round the cell π_i . This modification of the band \mathcal{T} decreases the number of H -cells in Φ , since one of the H -cells gets over the band \mathcal{T} .

The modified diagram can be non-minimal, but our surgery preserves q -bands and keeps the property that every q -band and every θ -band have at most one common (θ, q) -cells. So soon or later, this trick makes the inequality $\sum |\phi(z_i)|_a \leq 2$ true, and one can decrease \bar{x} , as was explained above. If one replaces the obtained diagram by a minimal one, then the condition (1) still holds since the maximal q -bands \mathcal{B}_1 and \mathcal{B}_{L-3} are completely determined by the boundary as this follows from Lemma 3.4. This contradicts to the assumption on the minimality of $|\bar{x}|$. Thus the condition (4) holds.

If Ψ' has H -cells in the subdiagram Γ'_i between the pair of k -bands \mathcal{B}'_i and \mathcal{B}'_{i+1} , which are not a pair of k_1 - and k_2 -bands, then these H -cells (with labels over the alphabet \mathcal{A}_1) cannot have common edges with the maximal θ -bands of Γ'_i since the θ -bands of Γ'_i must intersect either \mathcal{B}'_i or \mathcal{B}'_{i+1} by (4). This implies that Γ'_i has no H -cells at all because the path \bar{x} has no loops. The lemma is proved. \square

Lemma 4.9. *If a minimal diagram Δ has a hub, then the (cyclic shift of the) boundary path of Δ can be factorized as pp' so that the subpath p starts and ends with q -edges (and there is a simple path z in Δ with $z_- = p_-$, $z_+ = p_+$ such that the subdiagram bounded by the loop pz^{-1} has exactly one hub Π and the label $\phi(z)$ is equal in the group G_1 to a word of length $< |p|$).*

Proof. We may assume that a hub Π is chosen in Δ according to Lemma 3.11. Let $x'x''$ be the boundary path of Ψ as in Lemma 4.8. We will look for the path z in the minimal subdiagram Δ' obtained after removing of Ψ and Π from Δ . Therefore to prove the lemma, one may assume using the notation of Lemma 4.8, that $\Psi' = \Psi$ and $\bar{x} = x''$, i.e., the subdiagram Ψ itself has properties (3), (4), and (5) from Lemma 4.8. (We do not know if the diagram $\Psi' \cup \Pi \cup \Delta'$ is still minimal but we will use the minimality of Ψ' only.)

There is $d \geq 0$ such that there exist exactly d maximal θ -bands of Ψ crossing each of the k -bands $\mathcal{B}_1, \dots, \mathcal{B}_{L-3}$. This implies that the initial subbands $\mathcal{B}_i[d]$ of length d in all \mathcal{B}_i -s ($i = 1, \dots, L-3$) are copies of each other under the shifts of the indexes in their boundary labels.

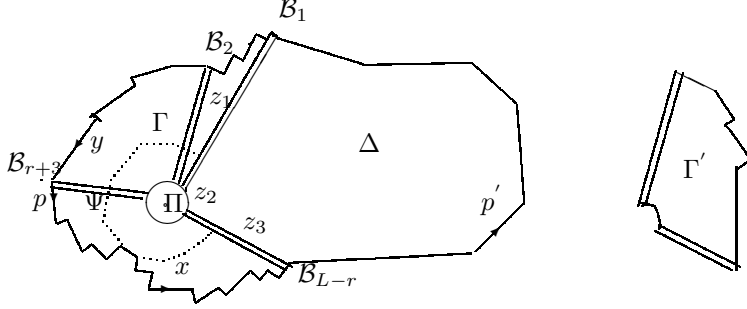
Let T be the set of remaining maximal θ -bands of Ψ , i.e., every band of T intersects exactly one of the bands $\mathcal{B}_1, \mathcal{B}_{L-3}$. It follows from Lemma 3.4 for Ψ that there is an integer l ($1 \leq l < L-3$) such that no θ -band of T crossing \mathcal{B}_1 crosses \mathcal{B}_{l+1} and no θ -band of T crossing \mathcal{B}_{L-3} crosses \mathcal{B}_l . We have either $(L-3)-l < (L-3)/2$ or $(l+1)-1 < (L-3)/2$ since L is even. Without loss of generality we choose the former inequality, and so $l \geq (L-2)/2$.

If \mathcal{B}_i is k_1 -band for some $i \leq 6$, then we will consider a smaller subdiagram bounded by \mathcal{B}_7 and \mathcal{B}_{L-3} instead of Ψ . (Respectively, we change the complimentary minimal subdiagram Δ' .) If none $\mathcal{B}_1, \dots, \mathcal{B}_6$ is a k_1 -band, then we does not change Ψ . Thus, in any case we can reindex the k -bands and assume that Ψ is bounded by \mathcal{B}_1 and \mathcal{B}_{L-r} for some $r \leq 9$ and that the bands $\mathcal{B}_1, \dots, \mathcal{B}_{r+3}$ are not k_1 -bands. Let them be $k_{i-}, \dots, k_{i \pm (r+2)}$ -bands for some i . We will assume that \mathcal{B}_{r+3} is a k_{i-r-2} -band.

Since $L \geq 40$, we have after such a reindexing that $l \geq (L-2)/2 - 6 > 12 \geq r+3$, and so no θ -band from the set T crossing the band \mathcal{B}_{L-3} crosses \mathcal{B}_{r+3} . We denote by Φ the part of the diagram Ψ bounded by \mathcal{B}_{r+3} and \mathcal{B}_1 . Let $\mathcal{T}_1^\Phi, \dots, \mathcal{T}_s^\Phi$ be the maximal θ -bands

of Φ . Then by the choice of the subdiagrams Ψ and Φ , every cell of Φ belongs to one of these θ -bands, and each of these bands crosses the k_i -band \mathcal{B}_1 . We will assume that \mathcal{T}_1^Φ is the closest band to the hub Π , and so on.

For every $j \geq 1$, we have that at least one of the two q -edges of every (θ, q) -cell of \mathcal{T}_i^Φ belongs to the top of \mathcal{T}_{i-1}^Φ (to $\partial\Pi$ if $i = 1$) because Ψ has no maximal q -bands except for the bands starting on Π . Assume that a band \mathcal{T}_i^Φ , starting with a (θ, q) -cell of \mathcal{B}_1 , terminates with an (θ, a) -cell π having no a -edges on $\partial\mathcal{T}_{i-1}^\Phi$. Then an a -edge and one θ -edge of Π lie on the boundary subpath x'' of Ψ , and so if one removes π from Ψ , the length of x'' does not increase since $|\pi|$ has 2 θ -edges and 2 a -edges, and all properties (3)-(5) hold for the remaining part of Ψ . Therefore we may assume that every edge f of $\text{tbot}(\mathcal{T}_i^\Phi)$ belongs to $\text{top}(\mathcal{T}_{i-1}^\Phi)$ (to $\partial\Pi$ for $i = 1$).



To construct the path z , we go along the side of the band \mathcal{B}_2 which is closer to \mathcal{B}_1 , then go along the part of the boundary of the hub which is not part of $\partial\Psi$, and finally go to $\partial\Delta$ along the side of \mathcal{B}_{L-r} which is closer to \mathcal{B}_{L-r+1} . Thus we have $z = z_1 z_2 z_3$ according to this definition, and respectively, $\phi(z) \equiv Z \equiv Z_1 Z_2 Z_3$. Let p be the subpath of $\partial\Delta$ and $\partial\Psi$ such that $p_- = z_-$, $p_+ = z_+$. Then the boundary path of Δ is of the form pp' , for an appropriate p' .

We denote by Γ the part of the diagram Φ bounded by \mathcal{B}_{r+3} and \mathcal{B}_2 . Let y be the common subpath of $\partial\Gamma$ and p with $y_- = p_- = z_-$. We may apply Lemma 4.4 to the pair (Γ, Φ) and obtain a new diagram Γ' over G_1 according to that lemma. One of the four boundary sections of Γ' is a subword $k_{i-1} \dots k_{i+r+1}$ of the word Σ_0 .

Note that \mathcal{B}_{L-r} is a k_{i+r+1} -band since we take the indexes of the k -letters modulo L . By Lemma 4.4, Γ' has a loop with label of the form $Z_1 Z_2 Z' Y'$, where Z' is the copy of the word written along the band \mathcal{B}_{r+3} , and $|Y'| \leq |y|$. Hence $Z = (Y')^{-1} (Z')^{-1} Z_3$ in G_1 . Therefore to prove that Z is equal in G_1 to a word of length $\leq |p|$, it suffices to prove that the word $(Z')^{-1} Z_3$ is equal in G_1 to a word of length $< |x|$, where $p = yx$.

We observe that the words Z' and Z_3 have equal prefixes of length d since $\mathcal{B}_{L-r}[d]$ is a copy of $\mathcal{B}_{r+3}[d]$. Therefore the word $(Z')^{-1} Z_3$ is equal to $(\bar{Z}')^{-1} \bar{Z}_3$, where the $(\bar{Z}')^{-1}$ copies the label of the part of the side of \mathcal{B}_{r+3} crossed by the θ -bands from the set T only, and \bar{Z}_3 is the label of the part of z_3 crosses by the θ -bands of T only. We denote the union of these two subsets of T by \bar{T} . The length of $(\bar{Z}')^{-1} \bar{Z}_3$ does not exceed the number $|\bar{T}|$ of bands in \bar{T} since neither of the bands from T crosses both \mathcal{B}_{r+3} and \mathcal{B}_{L-r} . By Lemma 3.4, every band of \bar{T} must end on the subpath x . Hence $|(\bar{Z}')^{-1} \bar{Z}_3| \leq |x|$. In

fact this inequality is strict since $L - r > r + 3$ (as $r \leq 9$) and so x must includes some q -edges as well. The lemma is proved. \square

5 Spaces of words

5.1 Spaces of boundary labels of some diagrams.

We call a disc *simple* if it has no H -cells and either all its θ -edges have labels from Θ or all of them have labels from $\hat{\Theta}$.

Lemma 5.1. *Let Δ be a diagram having exactly one hub Π . Then there is a diagram $\bar{\Delta}$ with the same boundary label as Δ such that $\bar{\Delta}$ has a simple disc subdiagram D , and the annular diagram $\Gamma = \bar{\Delta} \setminus D$ is a minimal annular diagram without θ -annuli.*

Moreover, one may assume that the boundary label of D is of the form $(k_1 W_1 k_2 W_2 \dots k_L W_L)^{\pm 1}$, where $k_1 W_1 k_2 W_2 \dots k_L W_L$ is accepted by either machine $\mathcal{S}(L)$ or by $\hat{\mathcal{S}}(L)$, and the lengths of θ -annuli in D do not exceed $N + L(\text{space}_{\mathcal{S} \cup \hat{\mathcal{S}}}(W_2))$.

Proof. We may assume that Δ is a minimal diagram. Let D_1 be a maximal disc subdiagram of Δ , and denote by $\mathcal{K}_1, \dots, \mathcal{K}_L$ the maximal k_1, \dots, k_L -bands of D_1 starting on the hub Π . We denote by Γ_i the maximal accepted i -sector of D_1 bounded by k_i and k_{i+1} ($i = 1, \dots, L$). By lemmas 3.8 and 3.9(1), these sectors have no H -cells for $i \neq 1$ and each of them is a copy or a mirror copy of the 2-sector Γ_2 .

Now we replace the 1-sector Γ_1 by the replica Γ'_2 of Γ_2 in D_1 . (For these aid, one can made a cut along $\mathcal{K}_1, \mathcal{K}_2$ and the part of the boundary of Π between these k -bands, and insert two mirror copies of Γ'_2 along this cut.) By the definition of replica, we obtain a modification D_2 of the disc diagram D_1 with boundary label of the form $k_1 W_1 k_2 W_2 \dots k_L W_L$, where W_1 is a mirror copy of W_2 or W_1 has no a -letters. Since $k_2 W_2 k_3$ is an accepted 2-sector word, the word $k_1 W_1 k_2 W_2 \dots k_L W_L$ is accepted by either machine $\mathcal{S}(L)$ or machine $\hat{\mathcal{S}}(L)$ by Lemma 3.1. Moreover the length of this computation does not exceed the length of the computation of $\mathcal{S} \cup \hat{\mathcal{S}}$ by the same lemma. Therefore by Lemma 3.9(2), the disc D_2 can be replaced by a disc D_3 which has no H -cells, whose labels of θ -edges either all belong to Θ or all belong to $\hat{\Theta}$, and whose number of θ -annuli does not exceed that number for D_2 .

Now, if necessary, the annular diagram $\Delta \setminus D_3$ can be replaced by a minimal diagram Γ over the group G_1 . Assume that Γ has a θ -annulus \mathcal{T} . By Lemma 3.4, \mathcal{T} surrounds the disc diagram D_3 , and so D_3 can be included in a larger disc subdiagram D_4 which contain more (k_i, θ) -cells for $i \neq 1, 2$ since the extensions of the \mathcal{K}_i -s have to cross \mathcal{T} . Then one can make the surgery as above and replace D_3 by a larger simple disc D_4 . This procedure terminates, because we do not change the number of (k_i, θ) -cells ($i \neq 1, 2$) in the compliment of disc when passing from D_1 to D_2 and from D_2 to D_3 and reduce this number when passing from D_3 to D_4 . (Recall that the rank of such cells is higher than the ranks of other cells in diagrams over G_1 .) Thus the procedure terminate with a desired diagram Γ .

Finally, one can replace the disc by a disc corresponding to a computation of minimal space and use Lemma 3.1 to make the second claim of the lemma true. \square

Lemma 5.2. *There are positive constants c_1 and c_2 with the following property. For the boundary label $w = k_1W_1 \dots k_tW_t$ of the simple disc D from lemma 5.1, there is a sequence of elementary transformations (say, simple sequence) $w = w_0 \rightarrow w_1 \rightarrow \dots \rightarrow w_t = 1$ using the relations of $G(\mathcal{S} \cup \hat{\mathcal{S}}, L)$ and the hub relation (3.6), such that $|w_i| \leq c_1 S'_S(|W_2|) + c_2$ ($i = 0, 1, \dots, t$).*

Proof. Let us start with the words $w_0 = w(0), w(1), \dots, w(m) = \Sigma_0$, written on the boundaries of the θ -annuli of D . By Lemma 5.1 and 3.1, there is $c_1 > 0$ such that $|w(i)| \leq c_1 S'_S(|W_2|) + N$. Notice that $w(i)$ is written on the top of a θ annulus \mathcal{T} of D and $w(i+1)$ is written on its bottom. The band \mathcal{T} has N (θ, q) -cells. For these cells the combinatorial lengths of tops and bottoms differ by at most ± 1 . The remaining cells are (θ, a) -cells, and there tops and bottoms have one a -edge. Therefore one can insert several elementary transformations between $w(i)$ and $w(i+1)$ corresponding to a sequential removal of the cells of \mathcal{T} so that the combinatorial length of the words obtained after the refinement of our sequence does not exceed $|w_i| + N + 2$. To complete the proof, it suffices to set $c_2 = 2N + 2$. \square

Lemma 5.3. *Assume that (1) a minimal diagram Δ has no hubs and no q -bands or (2) Δ is a union of a simple disc D and a minimal annular diagram Γ over G_1 surrounding the disc subdiagram D and having no θ -annuli, and every maximal q -band of Δ starts on the hub of D . Then the sum of perimeters σ_H of all H -cells in Δ does not exceed $c_3 |\partial\Delta|$ for some constant c_3 independent of Δ .*

Proof. We consider the condition (2) of the lemma only since a simplified argument works if Δ satisfies condition (1).

Two different H -cells cannot be connected by an a -band in a minimal diagram since otherwise this subdiagram can be replaced by a diagram with one H -cell and several (θ, a) -cells contrary to the minimality of the diagram. (See Lemma 3.12 (2) in [21].) A maximal a -band \mathcal{A} cannot connect a -edges of the same H -cell π in a disc subdiagram by Lemma 3.12 (3) in [21], and also \mathcal{A} and π cannot surround the disc D since their boundaries have no q -edges. Therefore every maximal a -band starting on an H -cell ends either on $\partial\Delta$ or on ∂D , or on a (θ, q) -cell of the annular diagram Γ . Therefore to estimate σ_H , we should give an estimate for the number $n_{\theta, q}$ of (θ, q) -cells in Γ and for the number of a -letters in the word W_1 , where $k_1W_1 \dots k_LW_L$ is the boundary label of D . (Recall that the edges of H -cells are labeled by a -letters from the alphabet \mathcal{A}_1 and so cannot be connected by a -bands with subpaths of ∂D labeled by W_2, \dots, W_L .)

Let \mathcal{T} be a maximal θ -band of Γ . Since both the start and the end θ -edges of \mathcal{T} must belong to $\partial\Delta$, it follows from Lemma 3.4 that \mathcal{T} crosses every q -band of Γ at most once, and so has at most N (θ, q) -cells, because every maximal q -band of Γ starts on the disc D . The number of maximal θ -bands of Γ does not exceed $|\partial\Delta|$ since Γ has no θ -annuli. It follows that $n_{\theta, q} \leq N |\partial\Delta|$.

Since we have $|W_1|_a \leq |W_2|_a = |W_3|_a = \dots$ for the simple disk D , we will look for an upper estimate for $|W_2|_a$. Every maximal a -band \mathcal{A} starting on the subpath p of ∂D labeled by W_2 cannot end on p by Lemma 3.4 since the word W_2 is reduced and has no θ -letters. Therefore \mathcal{A} terminates either on one of two closest maximal q -bands Q and Q' starting on D or on $\partial\Delta$. The lengths of Q and Q' are at most $|\partial\Delta|$ as this was explained in the previous paragraph, and so each of the sides of these q -bands has at most $|\partial\Delta|$ a -edges. Therefore $|W_2|_a \leq 3|\partial\Delta|$.

Thus, $\sigma_H \leq c_3 |\partial \Delta|$ for the constant $c_3 = 1 + 3 + 2N$. \square

Assume that a word w vanishes in the group G given by relations (3.4), (3.5), and (3.6). Then we denote by $Space_G(w)$ the minimal number m such that there is an elementary reduction of w to the empty word such that at every step i , we have a tuple of words $(w_{i1}, \dots, w_{i,s(i)})$ with $|w_{i1}| + \dots + |w_{i,s(i)}| \leq m$. For a set \mathcal{W} of words vanishing in G (call such a set *vanishing*), we define the space function $f_{G,\mathcal{W}}(x)$ as the maximum of $Space_G(w)$ over the words $w \in \mathcal{W}$ with $|w| \leq x$. (Thus we use the length $| \cdot |$ here unlike the length $|| \cdot ||$ used in Introduction.)

Now we denote by \mathcal{W}_1 the set of words read on the boundaries of simple discs, and we say that $w \in \mathcal{W}_2$ if w can be read on the boundary of an H -cell.

Lemma 5.4. *The function f_{G,\mathcal{W}_2} is bounded from above by a function equivalent to f_{G,\mathcal{W}_1} .*

Proof. Assume that $|w| = n \geq 1$ and $w \in \mathcal{W}_2$. Since the set of generators a_1, \dots, a_m of H is symmetric, for every $i \leq m$, we have a positive relation of the form $a_i a_{i'}$ for some $i' \leq m$. By Lemma 3.2 these 2-letter relations are consequences of the relations of G . Since the set of 2-letter relations is finite, there is a constant c_4 such that one can letter-for-letter convert w in a positive word u of the same length, and the space of the corresponding chain of transformations is $\leq n + c_4$. The word u is a product of cyclic shifts of the words $\Sigma(u)$ and $\hat{\Sigma}(u)$ as this was explained at the end of Subsection 3.1. Here both words $\Sigma(u)$ and $\hat{\Sigma}(u)$ belong to \mathcal{W}_1 and their lengths are at most $Ln + N \leq (L + N)n$. Therefore for any $w \in \mathcal{W}_2$, we have $space_G(w) \leq 2f_{G,\mathcal{W}_1}((L + N)n) + c_4$ which implies the statement of the lemma. \square

Lemma 5.5. *Let $C > 0$. Assume that for every w from a vanishing set $\in \mathcal{W}$, there is a sequence $w = w_0 \rightarrow w_1 \rightarrow \dots \rightarrow w_t = 1$ such that for every $i = 0, \dots, t - 1$, a cyclic shift of the word w_i is freely equal to a product of a cyclic shift of w_{i+1} and a word v_i from a vanishing set \mathcal{W}' , where $\max(|v_i|, |w_i|) \leq C|w|$ ($i = 0, \dots, t - 1$). Then the function $f_{G,\mathcal{W}}$ is bounded from above by a function equivalent to $f_{G,\mathcal{W}'}$.*

Proof. The condition of the lemma implies that we can apply the following series of elementary transformations to w_i . First series of transformations replaces the word by its cyclic shift, the second series deletes/inserts mutual inverse letters operating with the words of length $\leq 2C|w|$, then we split the obtained word in a product of a cyclic shift of w_{i+1} and the word v_i , then we keep w_{i+1} unchanged and use the appropriate procedure reducing the word v_i to the empty word, and finally obtain the word w_{i+1} using cyclic shifts. Clearly, we have that the space of this procedure is at most $2C|w| + f_{G,\mathcal{W}'}(C|w|)$. Thus, by induction on i , we have $Space_G(w) \leq 2Cn + f_{G,\mathcal{W}'}(Cn)$ for arbitrary word $w \in \mathcal{W}$ of length at most n . The lemma is proved. \square

Now we introduce the set \mathcal{W}_3 of the boundary labels of diagrams Δ satisfying the condition of Lemma 5.3, i.e., either (1) Δ is a minimal diagram Δ having no hubs and no q -bands or (2) Δ is a union of a simple disc D and a minimal annular diagram Γ over G_1 surrounding the disc subdiagram D and having no θ -annuli, and every maximal q -band of Δ starts on the hub of D .

Lemma 5.6. *The function f_{G,\mathcal{W}_3} is bounded from above by a function equivalent to $f_{G,\mathcal{W}_1 \cup \mathcal{W}_2}$.*

Proof. We will assume that a word w of length $n \geq \delta$ is the boundary label of a diagram Δ satisfying condition (2) in the definition of the set \mathcal{W}_3 . Every cell of the annular subdiagram Γ is either H -cell or a θ -cell. Therefore there is a sequence of diagrams $\Delta = \Delta_0, \Delta_1, \dots, \Delta_t = D$ such that for $i = 1, \dots, t$, the diagram Δ_i results from Δ_{i-1} after one cut off either (a) an H -cell or (b) a rim θ -band, or an edge e such that ee^{-1} belongs to the boundary path of Δ_{i-1} . The surgery of types (b) and (c) decreases the perimeter by Lemma 4.6. Although the surgery of type (a) can increase the perimeter, it follows from Lemma 5.3 that the perimeter of every diagram Δ_i is at most $(1 + c_4)n$.

Now we have a sequence $w = w_0, w_1, \dots, w_t, 1$ where w_0, \dots, w_t are the boundary labels of $\Delta_0, \Delta_1, \dots, \Delta_t = D$, of lengths at most $(1 + c_4)n$, such that, for every $i = 0, \dots, t$, a cyclic shift of the word w_i is a product of a cyclic shift of the word w_{i+1} and a word v_i , where v_i is either a boundary label of an H -cell of Δ or a 2-letter word aa^{-1} , or the boundary label of the simple disc D , or the boundary label of the rim θ -band of Δ_i . In all these cases $|v_i| \leq 2(1 + c_4)n$, in the case of rim band, we obviously have $\text{Space}_G|v_i| \leq 2(1 + c_4)n$, and in other cases $\text{Space}_G|v_i| \leq f_{G, \mathcal{W}_1 \cup \mathcal{W}_2}((1 + c_4)n)$. Therefore one can apply Lemma 5.5 and complete the proof. \square

By definition, the set of words \mathcal{W}_4 contains the set \mathcal{W}_3 and consists of boundary labels of diagrams Δ , where either (1) Δ is a minimal diagram having no hubs or (2) Δ is a union of a simple disc D and a minimal annular diagram Γ over G_1 surrounding the disc subdiagram D and having no θ -annuli.

Lemma 5.7. *The function f_{G, \mathcal{W}_4} is bounded from above by a function equivalent to f_{G, \mathcal{W}_3} .*

Proof. Again we assume that a word w of length $n \geq \delta$ is the boundary label of a diagram Δ satisfying condition (2) in the definition of the set \mathcal{W}_4 . Assume that Δ has a maximal q -band \mathcal{C} which does not start or terminate on the simple disc D . Then Δ is separated in 3 subdiagrams: Γ_1 contains the disc D , $\Gamma_2 = \mathcal{C}$, and Γ_3 is the remaining part of Δ . On the one hand, the lengths of the top and bottom of \mathcal{C} is equal to the number of cells m in \mathcal{C} since every cell of \mathcal{C} has one θ -edge and at most one a -edge on each of the sides of \mathcal{C} . On the other hand, each maximal θ -band of Δ crossing \mathcal{C} must start and terminate on $\partial\Delta$. This implies that the perimeters of each subdiagram Γ_1, Γ_2 , and Γ_3 are at most $2n$. (Here we use the definition of length and take into account that the band \mathcal{C} starts and ends on q -edges.)

Therefore there is a sequence of diagrams $\Delta = \Delta_0, \Delta_1, \dots, \Delta_t$ of perimeters $\leq 2n$ such that for $i = 1, \dots, t - 1$, the diagram Δ_i results from Δ_{i-1} after one cut off either (a) an subdiagram without q -bands or (b) a q -band, and Δ_t has no maximal q -bands except for those starting/terminating on the hub of D . Let $w = w_0, w_1, \dots, w_t$ be the boundary labels of these diagrams. Every word w_i ($i = 0, \dots, t$) of the series $w = w_0, w_1, \dots, w_t, w_{t+1} = 1$ (or its cyclic shift) is a product of a cyclic shift of the word w_{i+1} and a word v_i , where v_i either belongs to \mathcal{W}_3 or it is the boundary label of a q -band. In all these cases $|v_i| \leq 2n$, in the later case we obviously have $\text{Space}_G|v_i| \leq 2n$, and in former cases $\text{space}_G|v_i| \leq f_{G, \mathcal{W}_3}(2n)$. To complete the proof, we apply Lemma 5.5. \square

Lemma 5.8. *Let \mathcal{W}_5 be the set of all words vanishing in G . The space function $f_{G, \mathcal{W}_5}(n)$ of the group G is bounded from above by a function equivalent to $f_{G, \mathcal{W}_3 \cup \mathcal{W}_4}(n)$.*

Proof. Let $w = 1$ in G and $|w| = n > 0$. If $w = 1$ in G_1 , then $\text{Space}_G(w) \leq \text{Space}_{G, \mathcal{W}_3}(w)$ by Lemma 5.7. Otherwise the minimal diagram Δ with boundary label w has $t \geq 1$ hubs,

and by Lemma 4.9, a cyclic shift of the word $w = w_0$ is a product of a word v_1 written on the boundary of a diagram Γ having one hub, and a word w_1 which is a boundary label of a diagram with $t - 1$ hubs, and $|w_1| \leq n, |v_1| \leq 2n$. By Lemma 5.1, $v_1 \in \mathcal{W}_4$. Now induction on t gives a series of words $w_0, w_1, \dots, w_t = 1$ and words v_1, \dots, v_t satisfying the conditions of Lemma 5.5 with $C = 2$, and our statement follows from that lemma. \square

5.2 Proofs of main statements.

Proof of Theorem 1.2. Let a *DTM* with space function $f(n)$ recognize the language of vanishing in H words in the finite set of generators of the group H . By Lemma 2.10 (a), there is an S -machine \mathcal{S} recognizing the same language, and the generalized space function $S'_\mathcal{S}(n)$ of \mathcal{S} is equivalent to $f(n)$. The group G constructed on the basis of \mathcal{S} in Subsection 3.1 is finitely presented and contains H as a subgroup by Corollary 3.1. The consecutive application of lemmas 5.8, 5.7, 5.6, 5.4, and 5.2 results inequality $f_{G, \mathcal{W}_5}(n) \preceq S'_\mathcal{S}(n)$. Note that the space functions $f_{G, \mathcal{W}_5}(n)$ and $s_G(n)$ of G are equivalent since the lengths functions $\| * \|$ and $| * |$ satisfy inequalities $\delta \|w\| \leq |w| \leq \|w\|$ for every word w . Hence $s_G(n) \preceq S'_\mathcal{S}(n)$. Since $S'_\mathcal{S}(n) \sim f(n)$ by Lemma 2.10 (2), we have $s_G(n) \preceq f(n)$.

To invert this inequality, we first note that $s_G(n) \succeq \log d(n)$, where $d(n)$ is the Dehn function for G . (Indeed, up to equivalence, the length t of a rewriting $W_0 = (w_0) \rightarrow \dots \rightarrow W_t = ()$ without repetitions does not exceed $\exp(\max_{i=0}^t \|W_i\|)$; see also Theorem C in [7]). Therefore it suffices to show that $d(n) \succeq \exp(f(n))$. By lemma 2.4(3), $S'_{M'}(n) \sim f(n)$, and by Lemma 2.9 (4), $T'_\mathcal{S}(n) \succeq \exp(S'_{M'}(n))$. Thus it remains to explain that $d(n) \succeq T'_\mathcal{S}(n)$.

Let W be a word accepted by \mathcal{S} , such that $1 \leq |W|_a \leq n$ and $\text{time}_\mathcal{S}(W) = T'_\mathcal{S}(n)$. Then the word $V = k_1 W_1 \dots k_{N-1} W_N k_N$ (where W_i are copies or mirror copies of W) is accepted by $\mathcal{S}(L)$ and therefore it is conjugate of the word Σ_0 (see subsection 3.1). Hence $V = 1$ in G . Let Δ be a minimal diagram over G with boundary label V . Since $\partial\Delta$ has only one k_1 -edge, the maximal k_1 -band starting on this k_1 -edge must end on the boundary of a hub. Hence Δ has a hub Π satisfying to the condition of Lemma 3.11. If one removes Π together with the bands $\mathcal{B}_1, \dots, \mathcal{B}_{L-3}$ and with subdiagrams Γ_i ($i = 1, \dots, L-4$), then the remaining diagram Δ' has at most $L - (L-3) + 3 = 6$ k -edges. It follows from Lemma 3.11 that Δ' has no hubs since $6 < L-3$. Thus Δ has exactly one hub.

Obviously, every k_i -band starting on $\partial\Pi$ ends on $\partial\Delta$ ($i = 1, \dots, L$), and so we can consider the 2-sector Γ of Δ bounded by k_2 - and k_3 -bands. By Lemma 3.9(1), the number of maximal θ -bands of Γ is equal to the length of a computation of $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$ accepting the word W . By Lemma 3.1, we can replace $\mathcal{S}(L) \cup \hat{\mathcal{S}}(L)$ by \mathcal{S} in the previous phrase, and so $\text{area}(\Delta) > T'_\mathcal{S}(n)$. Since $\|V\| \leq C|W|_a$ for a constant C , we have $d(Cn) > T'_\mathcal{S}(n)$, and the required lower bound is obtained. \square

Proof of Theorem 1.4. Let M be a *DTM* with space function $f(n)$. Then as in the proof of Theorem 1.2, we can construct an S -machine \mathcal{S} with $S'_\mathcal{S}(n) \sim f(n)$. But now we simplify the construction of the group G : It is defined by the relations associated with the machine $\mathcal{S}(L)$ only and the hub relation (there is no $\hat{\mathcal{S}}(L)$ now, and so we have no H -relations at all). The (simplified) proof of Theorem 1.2 works in this simplified setting and therefore $s_G(n) \sim f(n)$. \square

Proof of Corollary 1.5. Let an *NTM* has an *FSC* space function $f(n)$ and solves the word problem in a finitely generated group H . Then by Savitch's Theorem (see [9],

Theorem 1.30), there is a *DTM* which solves the same problem with space $\sim f(n)^2$. It remains to refer to Theorem 1.2. \square

We need one more lemma to prove Corollary 1.6. This is a version of Savitch's theorem (see Theorem 1.30 in [9]), but instead of simulation of the work of a *NTM* by a *DTM*, now we need a *DTM* computing the space function of given *NTM*.

Lemma 5.9. *Let M be an *NTM* with space function $S(n)$ bounded from above by an *FSC* function $f(n)$. Then there is a *DTM* M_0 such that (1) M_0 computes $S(n)$, i.e., for any input $n \in \mathbb{N}$ given in binary, it computes $S(n)$; (2) the space function $S_{M_0}(n)$ is $O(f(n)^2)$.*

Proof. Without loss of generality, we may assume that M satisfies the \vec{s}_{10} condition.

If u is an accepted input word for M and $\|u\| \leq n$, then the time of any computation (without repetitions) of space $\leq f(n)$ accepting u is at most $\leq 2^{cf(n)}$, for some integer $c > 0$ since all configurations in this computation are of length $\leq f(n) + c_0$ for a constant c_0 . So the goal for a *DTM* M' we want to define, is to find a computation C of M of minimal space and of length at most $2^{cf(n)}$ which connects the input configuration and the accept configuration of M , and then to compute the space of this computation. (If such a computation exists; otherwise M' says that $u \notin \mathcal{L}_M$.) Indeed, the required machine M_0 will examine all words u with $\|u\| \leq n$ in lexicographical order, will switch on M' for every such input word u , and will compare the spaces $space_M(u)$ of u -s on an additional tape keeping only the maximal one after every return.

For arbitrary words w and w' and $k \in \mathbb{N}$, define predicates $reach_n(w, w', k)$ to mean that w and w' are configurations of M and there is a computation $w \rightarrow \dots w'$ with time $\leq k$ and with space $\leq f(n) + c_0$. By this definition, M accepts an input word u of combinatorial length $\leq n$ iff $reach_n(w_0, w_f, 2^{cf(n)})$, where $w_0 = w(u)$ is the unique input configuration on input u and w_f is the unique accept configuration of M .

Note that $reach_n(w, w', k + j)$ iff $(\exists w'')(reach_n(w, w'', k) \text{ and } reach_n(w'', w', j))$ and the minimal space of computations $w \rightarrow \dots \rightarrow w'$ is the minimum over all w'' of the maximums of minimal spaces for $w \rightarrow \dots \rightarrow w''$ and for $w'' \rightarrow \dots \rightarrow w'$. This observation leads to the following (slight) modification of Savitch's machine.

- Given $n \in \mathbb{N}$, compute $2^{cf(n)}$ (in binary) using that $f(n)$ is an *FSC* function.
- Then, $reach_n(w, w', 1)$ is true if $w \equiv w'$ or w transforms into w' under the application of a single command of M . The space of the computation $w \rightarrow w'$ is $\max(|w|_a, |w'|_a)$.
- If $k \geq 2$, then for all possible configurations w'' of M with length $\leq f(n) + c_0$, compute whether it is true that $reach_n(w, w'', [(k + 1)/2])$ and $reach_n(w'', w', [(k + 1)/2])$. Set $reach_n(w, w', k)$ to be true iff such w'' exists. Find the minimal space of computations $w \rightarrow \dots \rightarrow w'$ of length $\leq k$ using the information on the minimal space computations $w \rightarrow \dots \rightarrow w''$ and $w'' \rightarrow \dots \rightarrow w'$ of length $\leq [(k + 1)/2]$.

It is easy to see that the constructed machine M' computes, in particular, the space of any accepted input word u of length $\leq n$. Passing from k to $[(k + 1)/2]$, we need an additional space to store the information on k , w , w' , on the current w'' , and afterwards, on the minimal space of computations $w \rightarrow \dots \rightarrow w'$ of length $\leq k$. Clearly, this additional

space is $O(f(n))$, and since starting with $k = 2^{cf(n)}$, we divide k by $2 - cf(n)$ times, the total space used by M' and by M_0 is $O(f(n)^2)$. \square

Proof of Corollary 1.6. Assume that α is computable with space $\leq 2^{2^m}$. It follows that for $m = \lceil \log_2 \log_2 n \rceil$ we can recursively compute binary rationals α_m such that

$$|\alpha - \alpha_m| = O(2^{-m}) = O((\log_2 n)^{-1}) \quad (5.8)$$

and the space of the computation of α_m is at most n . In addition, one may assume that the number of digits in the binary expansion of α_m is $O(m)$. Therefore the computation of $\lceil \log_2 n \rceil$ (in binary) and of the product $\alpha_m \lceil \log_2 n \rceil$ needs space at most $O((\log_2 n)^2)$. Then we rewrite the binary presentation of $\lceil \alpha_m \lceil \log_2 n \rceil \rceil$ in unary (as a sequence of 1-s). This well-known rewriting (e.g., see p.352 in [25]) has the space function of the form $\lceil \alpha_m \lceil \log_2 n \rceil \rceil + O(1)$. One more rewriting of this type applied to the unary presentation of $\lceil \alpha_m \lceil \log_2 n \rceil \rceil$, will have the space function of the form $2^{\lceil \alpha_m \lceil \log_2 n \rceil \rceil} + O(1)$. Using (5.8), we can present this function as

$$2^{\alpha \lceil \log_2 n \rceil + O(1)} + O(1) \sim 2^{\alpha \lceil \log_2 n \rceil} \sim \lceil n^\alpha \rceil$$

Thus the subsequent application of the above mentioned *DTM*-s has space function equivalent to n^α , and we can apply Corollary 1.4 to obtain a finitely presented group with space function equivalent to n^α .

Now assume that a function $\lceil n^\alpha \rceil$ is equivalent to a space function of a finitely presented group G . Then by Proposition 1.1, there is an *NTM* M whose space function $S_M(n)$ is equivalent to $\lceil n^\alpha \rceil$, that is

$$c_1 n^\alpha < S_M(n) < c_2 n^\alpha \quad (5.9)$$

for some positive c_1 , positive integer c_2 , and every sufficiently large n . In particular, we have $S_M(n) < c_2 n^d$ for some integer d and every n . Since $c_2 n^d$ is an *FSC* function, we may apply Lemma 5.9, and obtain a *DTM* M_0 computing the function $S_M(n)$ with space $O(n^{2d})$. Hence M_0 computes the function $S_M(2^{2^m})$ of m with space $O((2^{2^m})^{2d})$. This space is less than $2^{2^{m+c_3}}$ for some c_3 . Hence for some $c_4 \in \mathbb{N}$, M_0 computes the function $S_M(2^{2^{m-c_4}})$ with space at most $2^{2^{m-1}}$.

Let us plug $n = 2^{2^{m-c_4}}$ to inequalities (5.9) and then take \log_2 of the terms. We obtain

$$\lambda_1 + \alpha 2^{m-c_4} \leq \log_2 S_M(2^{2^{m-c_4}}) \leq \lambda_2 + \alpha 2^{m-c_4}$$

where $\lambda_i = \log_2 c_i$ ($i = 1, 2$). It follows that

$$|\alpha - 2^{-m+c_4} \log_2 S_M(2^{2^{m-c_4}})| < c 2^{-m+c_4} = O(2^{-m}) \quad (5.10)$$

where $c = \max(|\lambda_1|, |\lambda_2|)$. Recall that $S_M(2^{2^{m-c_4}}) \leq 2^{2^{m-1}}$ and so this number has at most $2^{m-1} + 1$ binary digits. Therefore the real numbers $\log_2(S_M(2^{2^{m-c_4}}) 2^{-m+c_4})$ are computable with error $O(2^{-m})$ and space $O(2^{2^{m-1}})$. Now it follows from (5.10) that the real number α is computable with space 2^{2^m} . \square

References

- [1] G. Baumslag, A non-cyclic one-relator group all of whose finite quotients are cyclic. *J. Austral. Math. Soc.*, 10 (1969), 497-498.
- [2] J.-C. Birget, Time-complexity of the word problem for semigroups and the Higman embedding theorem, *Internat. J. Algebra Comput.* 8 (1998), 235-294.
- [3] J.-C. Birget, Functions on groups and computational complexity, *Internat. J. Algebra Comput.*, 14 (2004), no. 4, 409-429.
- [4] J.-C. Birget, J.-C., A. Yu. Olshanskii, E. Rips, M. V. Sapir, Isoperimetric functions of groups and computational complexity of the word problem. *Ann. of Math.* (2) 156 (2002), no. 2, 467-518.
- [5] N.Brady and M.Bridson, There is only one gap in the isoperimetric spectrum, *Geometric and Functional Analysis*, 10 (2000), 1053-1070.
- [6] N. Brady, T.Riley, and H.Short, The geometry of the word problem for finitely generated groups, *Advanced Courses in Mathematics*, CRM Barselona, Birkhauser-Verlag, Basel, 2007, x+206 p.p.
- [7] M.R. Bridson, T.R. Riley, Free and fragmenting filling length, *Journal of Algebra*, 307(1) (2007), 171-190.
- [8] D.E. Cohen, K.Madlener, and F.Otto, Separating the intrinsic complexity and the derivational complexity of the word problem for finitely presented groups, *Math. Logic Quart*, 39, no. 2 (1993), 143-157.
- [9] Ding-Zhu Du, Ker-I Ko, *Theory of Computational Complexity*, Wiley-Interscience Publ., N.Y.,2000, 512 p.p.
- [10] S. M. Gersten, Dehn functions and l_1 -norms of finite presentations. *Algorithms and Classification in Combinatorial Group Theory*, Springer, Berlin, 1992, 195-225.
- [11] S.M.Gersten, Isoperimetric and isodiametric functions. In G.Niblo and M.Roller editors, *Geometric group theory I*, Lecture Notes of LMS, 181, Camb. Univ. Press, 1993.
- [12] S.M.Gersten, T.R.Riley, Filling length in finitely presentable groups, *Geometriae Dedicata*, 92(1) (2002), 41-58.
- [13] M.Gromov, Hyperbolic groups, in: *Essays in Group Theory* (S.M.Gersten, ed.), M.S.R.I. Pub. 8, Springer, 1987, 75-263.
- [14] M.Gromov, Asymptotic invariants of infinite groups, in: *Geometric Group Theory. Vol. 2* (G.A.Niblo and M.A.Roller, eds.), London Math. Soc. Lecture Notes Ser., 182 (1993), 1-295.
- [15] LS R.C. Lyndon, P.E. Schupp, *Combinatorial Group Theory*, Springer-Verlag, 1977.
- [16] K.Madlener, F.Otto, Pseudo-natural algorithms for the word problem for finitely presented monoids and groups, *J. Symbolic Computation* 1(1985), 383-418.

- [17] A. G. Miasnikov, A. Ushakov, and Dong Wook Won, Word problem in Baumslag-Gersten group is polynomial time decidable, to appear.
- [18] A.Yu.Olshanskii, On the subgroup distortion in finitely presented groups, *Matem. Sbornik*, 188 (1997), N 11, 73-120 (in Russian).
- [19] A.Yu.Olshanskii and M.V.Sapir, Length and area functions in groups and quasi-isometric Higman embeddings, *Intern. J. Algebra and Comput.*, 11 (2001), no. 2 , 137-170.
- [20] A.Yu.Olshanskii and M.V.Sapir, Non-amenable finitely presented torsion-by-cyclic groups, *Publ. Math. IHES*, 96 (2003), no. 6, p.p. 43 - 169
- [21] A.Yu.Olshanskii and M.V.Sapir, Conjugacy problem and Higman embeddings, "Memoirs of the AMS" 170(2004), no. 804 p.p. vii+131
- [22] A.Yu.Olshanskii and M.V. Sapir, Groups with small Dehn functions and bipartite chord diagrams, *Geometric and Functional Analysis*, 16 (2006), 1324-1376
- [23] A.N.Platonov, Isoperimetric function of the Baumslag-Gersten group, (Russian) *Vestnik Moskov. Univ. Ser. I Mat. Mekh.* (2004), pp. 1217.
- [24] J.Rotman, An introduction to the theory of groups, 3d edition, Allyn and Bacon Inc., Boston, Mass, 1984.
- [25] M. V. Sapir, J. C. Birget, E. Rips, Isoperimetric and isodiametric functions of groups, *Annals of Mathematics*, 157, 2(2002), 345-466.